A Color Filter Array Interpolation Method Based on Sampling Theory

A Thesis
Presented to
The Academic Faculty

by

John W. Glotzbach

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2004

A Color Filter Array Interpolation Method Based on Sampling Theory

Approved by:

Ronald W. Schafer, Advisor

Russell M. Mersereau

Yucel Altunbasak

Date Approved: July 27, 2004

ACKNOWLEDGEMENTS

I would like to thank every person who has offered their help and support to me and to this research. This work would not have been possible without the help of my friends, family, and the DSP faculty at Georgia Tech. Thank you to each of the professors that taught the courses I had. Every signal processing course was interesting, informative, and inspirational. To my fellow students in CSIP, your technical and theoretical advice and interesting conversations have pushed me through this time. I would especially like to thank Amer Abufadel, Greg Slabaugh, Jordan Rosenthal, Raviv Raich, Majid Fozunbal, and Kevin Chan.

Guiding my research along the way was Dr. Ronald Schafer. I would like to thank you for all of the advice you have passed along and for all of the careful readings of this work and prior works. The group of people at Texas Instruments raised many good questions about this work and offered ideas that helped guide me with my research. Of these, I would especially like to thank Raj Talluri, Youngjun Yoo, and Klaus Illgner for all of the guidance they have offered.

To my family and friends that have offered their support and interest in my work, thank you. Your support has been invaluable. For anybody who has listened to me think out my problem of the day during a run, thanks for offering an ear. I'd especially like to thank Randy Roberts, Roy Furbank, and Jim Harper, who have heard more than their share of these problems. I would like to thank my parents whose unending love and support has been a huge motivation for me to continue this work. Thank you for all of your time and effort guiding me. Most of all, I would like to thank my wife, Tara, who has been there from the start of this work. It is for you that this work has been completed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

To reduce space and cost, many digital cameras use a single image sensor array with a color filter array to measure a color image. Instead of measuring a red, green, and blue value at every pixel, these cameras have a color filter built onto each pixel so that only one portion of the visible spectrum is measured. To generate a full-color image, the camera must estimate the missing two values at every pixel. This process is known as color filter array interpolation and is an important step in the image processing chain in digital cameras.

Many arrangements of color filter arrays exist, but the Bayer CFA is commonly used. The Bayer CFA samples the green image on half of the pixels of the imaging sensor on a quincunx grid. The other half of the pixels measure the red and blue images equally. Both the red and blue images are measured on interleaved rectangular sampling grids. This thesis deals exclusively with the Bayer CFA.

This thesis analyzes this problem with sampling theory. The red and blue images are sampled at half the rate of the green image. Thus, the red and blue images have a higher probability of aliasing in the output image. This is apparent when simple interpolation algorithms like bilinear interpolation are used for CFA interpolation. The green image is also subject to aliasing if the input image is not sufficiently band-limited.

Two reference algorithms, a projections onto convex sets (POCS) algorithm and an edge-directed algorithm by Adams and Hamilton (AH), are studied. Both algorithms address aliasing in the green image. Because of the high correlation among the red, green, and blue images, information from the red and blue images can be used to better interpolate the green image. The reference algorithms are studied to learn how this information is used. This leads to two new interpolation algorithms for the green image.

The red and blue interpolation algorithm of AH is also studied to determine how the inter-image correlation is used when interpolating these images. This study shows that

because the green image is sampled at a higher rate, it retains much of the high-frequency information in the original image. This information is used to estimate aliasing in the red and blue images. We present a general algorithm based on the AH algorithm to interpolate the red and blue images. This algorithm is able to provide results that are on average, better than both reference algorithms, POCS and AH.

CHAPTER I

INTRODUCTION

A color image measures the amount of light in each part of the visible spectrum for every pixel. A typical computer color image records three values for every pixel: red, green, and blue (RGB). A digital camera recording a color image to represent a scene must measure a red, green, and blue value at each pixel to fully represent the scene. This could be accomplished with three separate image sensor arrays with supporting optics to align the sensors. However, this would be expensive to implement so most consumer cameras simplify this system by using a single image sensor. To measure the red, green, and blue values with a single sensor array, the camera has a color filter built onto each pixel of the sensor array. Some of the pixels measure a red value, some measure a green value, and the rest measure a blue value. This filter arrangement is known as a color filter array (CFA).

The most common array used in digital cameras is the Bayer CFA [7], shown in Figure 1. The Bayer CFA measures the green image on a quincunx grid as shown in Figure 1(b).

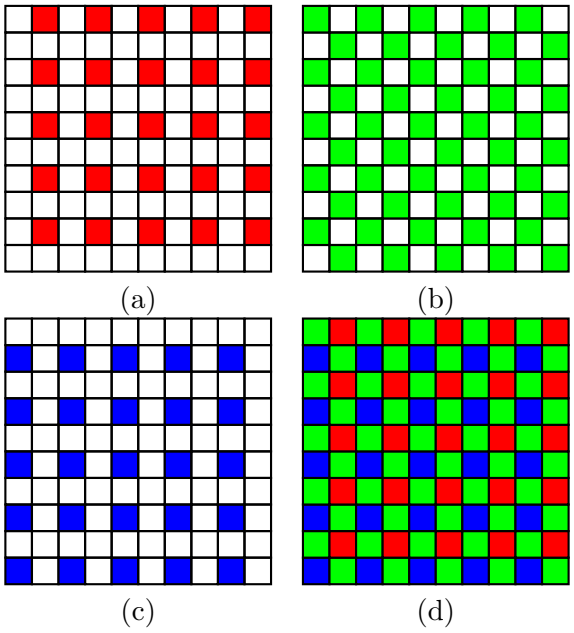


Figure 1: Bayer color filter array.

The green image is measured at half of the pixels. The remaining pixels are divided evenly between red and blue pixels. The red and blue images are measured on rectangular sampling grids that interlace with the green sampling grid. Because the green portion of the spectrum most closely matches the perceptual response of the human visual system [26, 27, 20], the green image is measured at a higher rate.

The problem with this arrangement is that no one pixel measures all three of the values needed to generate a color image. Therefore, a system is required to compute the missing data. This process is known as demosaicking or color filter array interpolation in the literature. Although CFA interpolation is just one step in a long image processing pipeline in a digital camera, it is a very important step in the production of a digital color image [5]. The output of the CFA interpolation process can affect the overall sharpness of the image and introduce undesired visual artifacts into the image. Aliasing is one such artifact. As seen in Figure 1, the red, green, and blue images are undersampled with respect to the sampling rate of the full image. Images with sharp edges or rapidly varying features contain components that are undersampled in all three images by the Bayer pattern, resulting in color aliasing artifacts.

1.1 Basic Approach

A simple approach to compute a color image from the color filter array samples is to use bilinear interpolation. Bilinear interpolation estimates each missing color value with the average of the nearest neighbors of the same color. This is illustrated in Figure 2. Pixels where a green value is missing have four neighboring green pixels. The average value of the neighboring pixels is computed and used to fill in the missing value. For the red and blue images, three cases are needed. In the first case, a missing red value has two neighboring red values horizontally, which are averaged to give the missing value. To fill in the missing value for the second case, the two neighboring pixels along the same column are averaged. The third case has four neighboring red pixels located diagonally adjacent to the missing red value. The same procedure is carried out to fill in the missing blue values. This interpolation algorithm is very simple, but it does not give good results. Figure 3 shows the

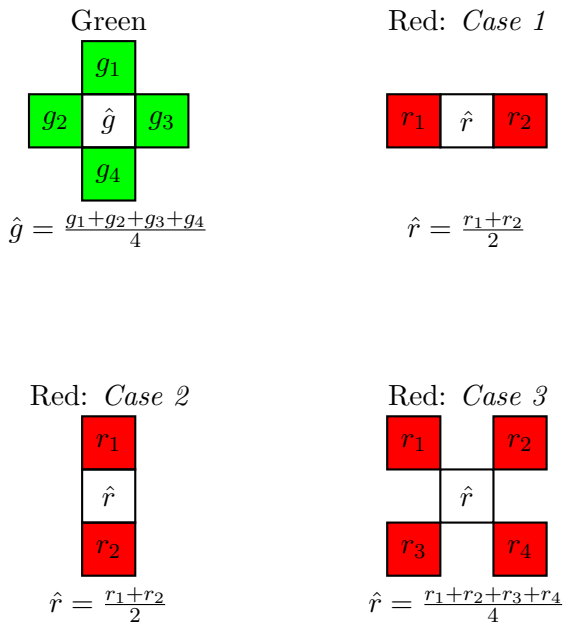


Figure 2: Illustration of bilinear interpolation.

Lighthouse image reconstructed using bilinear interpolation. This image contains several features that make it a difficult image for this problem. The sharp edges along the sides of the lighthouse and the rails of the lighthouse provide a good test for the edge response of the system. The siding along the buildings in the image contains edges at a regular frequency, which lead to aliasing artifacts due to undersampling of the image. The fence provides a chirp-like signal to test a system with high-frequency input signals. The original image, shown in Figure 3(a), was downsampled according to the Bayer CFA and reconstructed using bilinear interpolation. The reconstructed image is blurred compared to the original image and significant color artifacts are present. These artifacts are caused by aliasing in the red and blue images. This will be clear after Chapter 2 where sampling and aliasing issues are discussed.

A second approach is to extend the idea of cubic interpolation [16, 22] from grayscale image processing to CFA interpolation. Cubic interpolation fits a cubic polynomial to the data points in a 4x4 grid and computes the missing value by evaluating the polynomial at the point of the missing value. Because the red and blue images are sampled on rectangular grids, reconstructing these images is the same as increasing the image size by a factor of two in both directions. One way to apply bicubic interpolation to the green image is to

(a) (b)

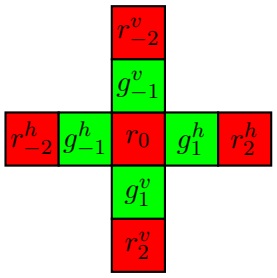
Figure 3: Bilinear interpolation results. (a) Original image. (b) Bilinear interpolation.

(a) (b)

Figure 4: Bicubic and spline interpolation results. (a) Bicubic interpolation result. (b) Spline interpolation result.

treat the quincunx grid as the union of two rectangular grids. The image is divided into two rectangularly sampled images and both are interpolated to the full image size. At the missing pixels, the two images are averaged to give the final interpolated image.

The result using bicubic interpolation is shown in Figure 4. Comparing this result with the result of bilinear interpolation, we can see that, with respect to aliasing in the output image, bicubic interpolation does not improve upon bilinear interpolation. Also shown is the result using spline interpolation [14, 18, 25], another standard interpolation technique. Neither technique showed much improvement over bilinear interpolation. The aliasing artifacts are objectionable in both images. In an application where undersampling is not present, bicubic interpolation would outperform bilinear interpolation, but in this case, any interpolation improvements are not seen because of the apparent aliasing artifacts.



Classifiers:

$$C_h = 2\left|g^h_{-1} - g^h_1\right| + \left|2r_0 - r^h_{-2} - r^h_2\right|$$

$$C_v = 2\left|g^v_{-1} - g^v_1\right| + \left|2r_0 - r^v_{-2} - r^v_2\right|$$

Interpolation:
If $C_h < C_v$ *then*

$$\hat{g} = \frac{1}{2}\left(g^h_{-1} + g^h_1\right) + \left(r_0 - \frac{1}{2}r^h_2 - \frac{1}{2}r^h_{-2}\right)$$

Else

$$\hat{g} = \frac{1}{2}\left(g^v_{-1} + g^v_1\right) + \left(r_0 - \frac{1}{2}r^v_2 - \frac{1}{2}r^v_{-2}\right)$$

Figure 5: Edge-directed interpolation method defined by Adams and Hamilton.

1.2 Adaptive Approach

One way to improve the interpolation results is to allow the algorithm to adapt to local features in the image. For example, if a horizontal edge is present in an image, then an interpolator would only want to use pixels along that edge for the interpolation and would not want to use pixels across the edge. For example, in the *Lighthouse* image, along the fence, a vertical interpolator would be used but along the side of the building, a horizontal interpolator would be used.

Adams and Hamilton have suggested one edge-directed algorithm [13, 3, 4, 1, 2]. To determine the interpolation direction, two classifiers are computed at each pixel. A horizontal classifier detects the amount of change in the horizontal direction and the vertical classifier measures the amount of change in the vertical direction. The direction with the smaller classifier is chosen as the interpolation direction. This algorithm is demonstrated in Figure 5. The classifiers, C_h and C_v, measure the amount of change in the horizontal and vertical directions, respectively. The classifiers are composed of the absolute difference between the neighboring green values and the central difference between the center red value and the closest two red values along the row or column. Both red and green differences are measured since both red and green images can give an indication of an edge. Once the

Figure 6: Edge-directed interpolation applied to *Lighthouse*.

classifiers are computed and an interpolation direction is chosen, the missing green value is computed. The neighboring green values along the interpolation direction are averaged. To this average value, a correction term from the red or blue image is added. The formulas are given in Figure 5. This algorithm is discussed in detail in Chapter 3.

Figure 6 shows the result image using the edge-directed interpolation algorithm defined by Adams and Hamilton (AH) on the *Lighthouse* image. Clearly, this method shows a large improvement over the standard interpolation methods. Along the fence, there are few aliasing artifacts and the edges in the image are sharp.

1.3 Thesis Outline

This thesis begins by analyzing the sampling system of the color filter array in Chapter 2. This provides a description of aliasing in the system and shows the difference among the

sampling of the red, green, and blue images. This analysis is used to provide a mathematical description of two reference algorithms, AH and POCS. Chapter 3 presents this description and shows how aliasing is handled by these two algorithms. Chapter 4 and Chapter 5 introduce new algorithms to perform CFA interpolation. Chapter 4 describes two methods to interpolate the green image. One method interpolates the green image with a linear, diamond-shaped filter, and then corrects this interpolation with correction filters applied to the red and blue images. The second method uses a filterbank system to perform the interpolation. Chapter 5 introduces two methods to interpolate the red and blue images such that aliasing is cancelled using information from the green image. In Chapter 6, eight algorithms are compared using a database of color images. Results of this comparison are presented. Chapter 7 lists the contributions of this thesis and presents future research ideas related to CFA interpolation.

CHAPTER II

SAMPLING EQUATIONS

The Bayer color filter array samples a color image by building red, green, and blue filters onto the imaging sensor in the camera system. In a sampling sense, this system interleaves a rectangular sampling grid for both the red and blue images and a quincunx sampling grid for the green image. Figure 7 shows the sampling grids for the red, green, and blue images. These grids are shown with respect to the full sampling grid of the full-size image. In the CFA interpolation problem, the open circles represent the missing values and must be interpolated to complete the full-color image.

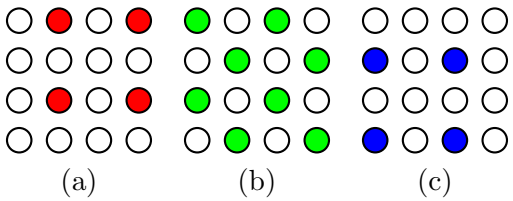


Figure 7: Bayer CFA sampling grids. (a) Red. (b) Green. (c) Blue.

The goal of this chapter is to derive the sampling equations to describe this system. We begin by deriving the equations for a signal sampled on a rectangular grid. Because the red and blue sampling grids are offset from each other, the effect of introducing a phase shift into the sampling grid is considered. This will give us the equations for the red and blue sampled signals. We then derive the equation for a signal sampled on a quincunx grid, giving us the sampling equation for the green image. A frequency domain illustration is given, describing how signals are affected by the rectangular and quincunx sampling grids.

In this chapter, it is assumed that samples are stored at the correct positions in the sampling grid and that color values not measured by the sensor are stored as zeros. The coordinate system is relative to the rectangular sampling grid of the full-size image and the size of the sensor is the same as the size of the output image. All of the sampled images also match the sensor size.

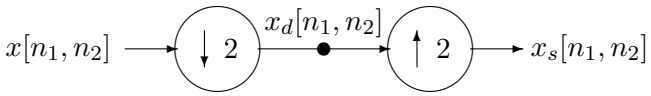


Figure 8: CFA rectangular sampling block diagram.

2.1 Rectangular Sampling Grid Without Phase Shift

Figure 8 shows the block diagram for the rectangular sampling system in the case where the sampling grid has no phase shift. This describes the system for

$$x_s[n_1, n_2] = \begin{cases} x[n_1, n_2] & n_1, n_2 \text{ even} \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

Multidimensional signal processing gives us two Fourier transform properties that are useful in deriving the sampling equation [11]. If

$$x_d[n_1, n_2] = x[Kn_1, Ln_2]$$

then

$$X_d(e^{j\omega_1}, e^{j\omega_2}) = \frac{1}{KL} \sum_{i_1=0}^{K-1} \sum_{i_2=0}^{L-1} X(e^{j\omega_1/K - 2\pi i_1/K}, e^{j\omega_2/L - 2\pi i_2/L}) \tag{2}$$

and if

$$x_u[n_1, n_2] = \begin{cases} x[n_1/K, n_2/L] & n_1 = iK, n_2 = jL \text{ where } i, j \text{ are integers} \\ 0 & \text{otherwise} \end{cases}$$

then

$$X_u(e^{j\omega_1}, e^{j\omega_2}) = X(e^{jK\omega_1}, e^{jL\omega_2}). \tag{3}$$

Using the property in (2) and noting that $K = L = 2$ in the Bayer CFA, it follows that

$$\begin{aligned} X_d\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}X\left(e^{j(\omega_1/2)}, e^{j(\omega_2/2)}\right) + \frac{1}{4}X\left(e^{j(\omega_1/2-\pi)}, e^{j(\omega_2/2)}\right) + \\ &\quad \frac{1}{4}X\left(e^{j(\omega_1/2)}, e^{j(\omega_2/2-\pi)}\right) + \frac{1}{4}X\left(e^{j(\omega_1/2-\pi)}, e^{j(\omega_2/2-\pi)}\right). \end{aligned} \tag{4}$$

Applying (3), we obtain the final sampling equation for the rectangular sampling grid with no phase shift. Since $X_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = X_d\left(e^{j2\omega_1}, e^{j2\omega_2}\right)$, the sampling equation for the rectangular sampling grid is given by

$$\begin{aligned} X_s\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}X\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{5}$$

2.2 Shifted Rectangular Sampling Grid

The rectangular sampling grid could be in one of four positions as shown in Figure 9. Figure 9(a) is the system described by (5), Figure 9(b) matches the red sampling in the Bayer CFA and Figure 9(c) matches the blue sampling.

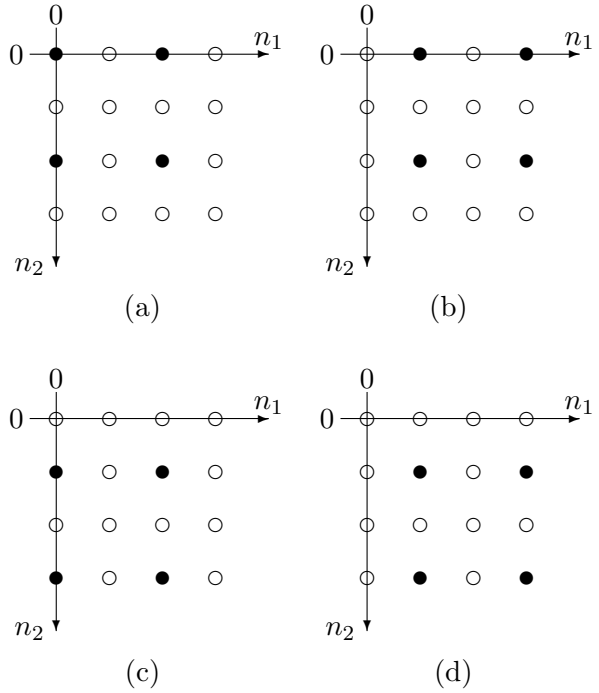


Figure 9: Possible phase positions for the rectangular sampling grid.

In Figure 9(b), the grid is shifted horizontally by one position. This system differs from the previous system by advancing the signal one position to the left before downsampling and delaying the upsampled signal one position to the right. This is shown in Figure 10.

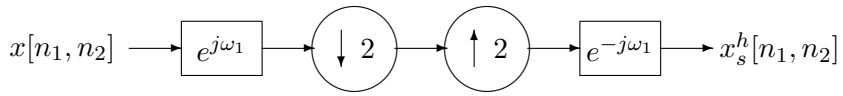


Figure 10: Block diagram for rectangular sampling with a horizontal phase shift.

The input to the downsampling block is now $e^{j\omega_1}X(e^{j\omega_1}, e^{j\omega_2})$. Downsampling by 2 creates four spectral copies of this signal.

$$\begin{aligned} X_d\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}e^{j\omega_1/2}X\left(e^{j(\omega_1/2)}, e^{j(\omega_2/2)}\right) + \frac{1}{4}e^{j(\omega_1/2-\pi)}X\left(e^{j(\omega_1/2-\pi)}, e^{j(\omega_2/2)}\right) + \\ &\quad \frac{1}{4}e^{j\omega_1/2}X\left(e^{j(\omega_1/2)}, e^{j(\omega_2/2-\pi)}\right) + \end{aligned}$$

$$\frac{1}{4}e^{j(\omega_1/2-\pi)}X\left(e^{j(\omega_1/2-\pi)},e^{j(\omega_2/2-\pi)}\right). \tag{6}$$

In the second and fourth terms, the factor $e^{j(\omega_1/2-\pi)}$ can be simplified to $e^{j\omega_1/2}e^{-j\pi} = -e^{j\omega_1/2}$. Therefore, the downsampled signal is described by

$$\begin{aligned} X_d\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}e^{j\omega_1/2}X\left(e^{j(\omega_1/2)},e^{j(\omega_2/2)}\right)-\frac{1}{4}e^{j\omega_1/2}X\left(e^{j(\omega_1/2-\pi)},e^{j(\omega_2/2)}\right)+ \\ &\quad \frac{1}{4}e^{j\omega_1/2}X\left(e^{j(\omega_1/2)},e^{j(\omega_2/2-\pi)}\right)- \\ &\quad \frac{1}{4}e^{j\omega_1/2}X\left(e^{j(\omega_1/2-\pi)},e^{j(\omega_2/2-\pi)}\right). \end{aligned} \tag{7}$$

The difference between this and the non-shifted version is the negative phase on the second and fourth terms. The upsampled signal is

$$\begin{aligned} X_u\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}e^{j\omega_1}X\left(e^{j\omega_1},e^{j\omega_2}\right)-\frac{1}{4}e^{j\omega_1}X\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)+ \\ &\quad \frac{1}{4}e^{j\omega_1}X\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)-\frac{1}{4}e^{j\omega_1}X\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{8}$$

By shifting to the right one sample at the high sampling rate, the upsampled signal is multiplied by $e^{-j\omega_1}$ to give the final sampling equation,

$$\begin{aligned} X_s^h\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}X\left(e^{j\omega_1},e^{j\omega_2}\right)-\frac{1}{4}X\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)+ \\ &\quad \frac{1}{4}X\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)-\frac{1}{4}X\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{9}$$

In the case of a vertical shift of the sampling grid, as in Figure 9(c), the delay factor is now $e^{-j\omega_2}$. This system is shown in Figure 11. The analysis used for the horizontal shift can be applied to this system to derive the final sampling equation,

$$\begin{aligned} X_s^v\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}X\left(e^{j\omega_1},e^{j\omega_2}\right)+\frac{1}{4}X\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)- \\ &\quad \frac{1}{4}X\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)-\frac{1}{4}X\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{10}$$

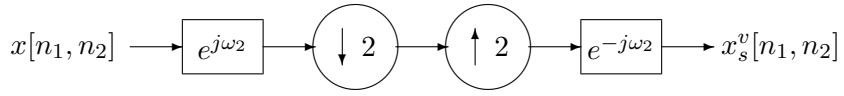


Figure 11: Block diagram for rectangular sampling with a vertical phase shift.

The final case involves a shift of the sampling grid by both one horizontal unit and one vertical unit, shown in Figure 9(d). In this case, the delay factor is the product of the

horizontal delay and the vertical delay, $e^{-j\omega_1}e^{-j\omega_2}$. The block diagram for this system is shown in Figure 12 and the final sampling equation is given by

$$\begin{aligned} X_s^d\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}X\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{11}$$

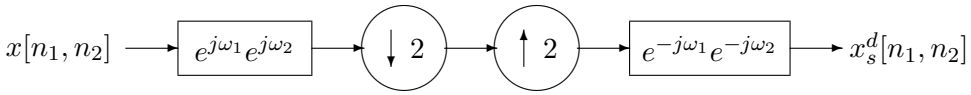


Figure 12: Block diagram for rectangular sampling with a diagonal phase shift.

2.3 Quincunx Sampling Grid

The quincunx sampling grid is used to sample the green image in the Bayer CFA. In Figure 13, the quincunx grid is shown as a sum of two rectangular sampling grids. This allows an

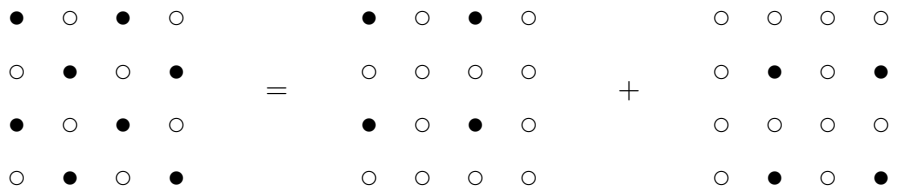


Figure 13: The quincunx sampling grid is the sum of two rectangular sampling grids.

easy derivation of the sampling equation,

$$x_q[n_1, n_2] = x_s[n_1, n_2] + x_s^d[n_1, n_2], \tag{12}$$

which gives the Fourier-domain representation,

$$\begin{aligned} X_q\left(e^{j\omega_1}, e^{j\omega_2}\right) &= X_s\left(e^{j\omega_1}, e^{j\omega_2}\right) + X_s^d\left(e^{j\omega_1}, e^{j\omega_2}\right) & (13) \\ &= \frac{1}{4}X\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) + \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) & (14) \\ X_q\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{2}X\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{2}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). & (15) \end{aligned}$$

2.4 Shifted Quincunx Sampling Grid

The quincunx grid could also be shifted by one horizontal unit, but it is still the sum of two rectangular sampling grids as shown in Figure 14. The time-domain representation is

$$x_q^h[n_1, n_2] = x_s^h[n_1, n_2] + x_s^v[n_1, n_2]. \tag{16}$$

In the frequency domain, this is written as

$$\begin{aligned} X_q^h\left(e^{j\omega_1}, e^{j\omega_2}\right) &= X_s^h\left(e^{j\omega_1}, e^{j\omega_2}\right) + X_s^v\left(e^{j\omega_1}, e^{j\omega_2}\right) && (17) \\ &= \frac{1}{4}X\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) + \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - \\ &\quad \frac{1}{4}X\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) && (18) \\ X_q^h\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{2}X\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{2}X\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) && (19) \end{aligned}$$

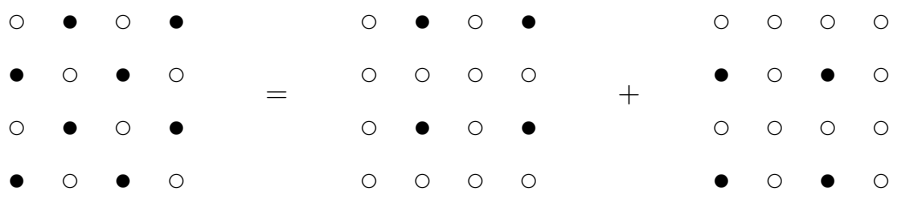


Figure 14: The quincunx sampling grid can be shifted by one horizontal unit, but it is still a sum of two rectangular grids.

2.5 Bayer CFA Sampling Equations

With this analysis, sampling equations can be given for the sampled red, green, and blue images in the Bayer CFA. The red image, sampled by a rectangular grid shifted by one horizontal unit, is expressed as

$$\begin{aligned} R_s\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}R\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{4}R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + \\ &\quad \frac{1}{4}R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{20}$$

The green image is sampled by a quincunx grid with no shift. Its sampling equation is

$$G_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}G\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{2}G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). \tag{21}$$

The blue image is sampled with a rectangular grid with a vertical shift. Therefore, it is expressed as

$$B_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{4}B\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{4}B\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - \frac{1}{4}B\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}B\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). \quad (22)$$

2.6 Interpretation in the Frequency Domain

Three sampling grids have been described. The full-size digital image is sampled on a rectangular sampling grid at the rate of the full sensor. With a CFA, the three images are sampled at lower rates. The red and blue images are also sampled on rectangular grids, but at only 1/4 of the full rate. This will be referred to as the 1/4-rate grid. The green image is sampled on a quincunx grid at 1/2 of the full rate, so this is denoted the 1/2-rate grid.

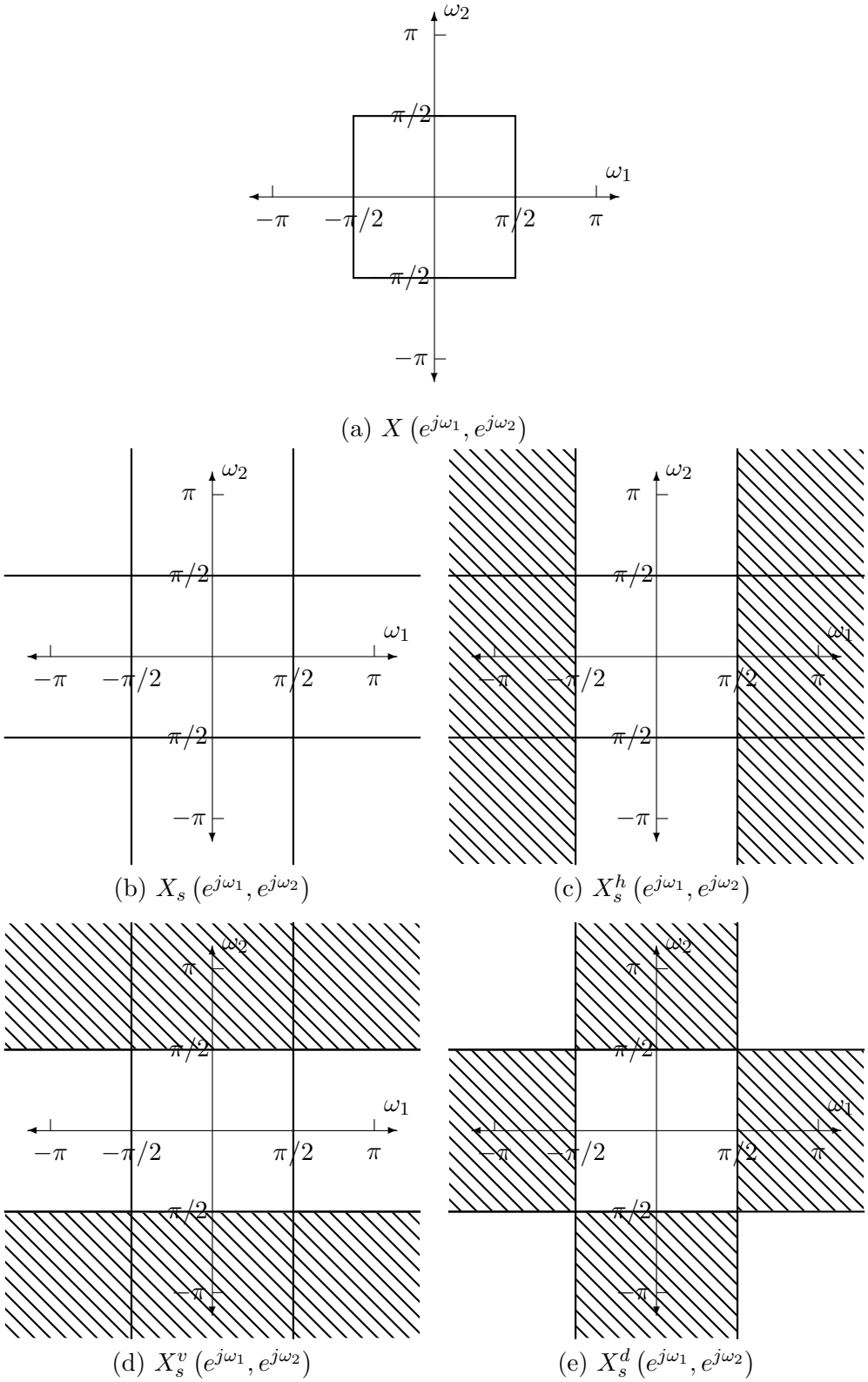
When an image is sampled with a rectangular grid, in the frequency domain, the original spectrum is copied at periodic locations in the frequency space. In order to avoid aliasing, the original image must have all of its energy contained within the Nyquist region for the 1/4-rate grid, a square for the case of rectangular sampling. This is shown in Figure 15, where the digital frequencies are with respect to the full sampling rate. In this figure, the original spectrum (i.e. the discrete spectrum when sampled at the full image rate) is shown in (a). Figure 15(b)-(e) show the cases of the four possible locations of the sampling grids. Figure 15(b) shows the case of rectangular sampling with no offset in the sampling grid. All of the copies of the original spectrum have the same phase as the original. In Figure 15(c), the sampling grid has a horizontal shift, so the spectral copies that are shifted horizontally have a phase inversion. The spectral copies in Figure 15(d) with a vertical shift have inverted phase because this sampling grid has a vertical offset. For the case of the sampling grid with both a horizontal and vertical offset, Figure 15(e) shows that the spectral copies with a horizontal shift or a vertical shift, but not both, are inverted. Compare the drawings in (c)-(e) with the expressions in (9)-(11).

The quincunx sampling grid also replicates the original spectrum at periodic locations. Because it is sampled at 1/2 the full rate, the original spectrum is replicated at only one other position, $(\omega_1, \omega_2) = (\pi, \pi)$. Instead of a rectangular Nyquist region, this sampling grid gives

a diamond-shaped Nyquist region, allowing the original image to have high horizontal and vertical frequencies present without aliasing. The diamond-shaped Nyquist region for the 1/2-rate grid is shown in Figure 16(a). In Figure 16(b), the frequency domain representation of the quincunx sampled grid is shown. This case has no phase shift in the sampling grid. For the quincunx sampling grid with a phase shift, the frequency-domain representation is shown in Figure 16(c). The phase shift leads to an inversion of the shifted copies in the frequency domain (i.e. negative copies).

Cameraman is a good example to illustrate this because it has several line features at different orientations. The example image is shown in Figure 17(a). The line features in the image correspond to lines in the spectrum, as seen in the original spectrum, shown in Figure 17(b). The sampled spectra for the rectangular and quincunx sampling grids are shown in Figure 17(c) and (d), respectively. These figures show the replication of the original spectrum throughout the frequency space.

For this example, it can also be seen that aliasing is a problem. The lines in the spectrum extend outside of the Nyquist region for both sampling grids. In the spectra for the sampled images, the replicated copies of these lines extend into the low-frequency regions of the spectrum, making it impossible for a general filter to separate the original spectrum from the spectral copies.



(a) $X\left(e^{j\omega_1}, e^{j\omega_2}\right)$

(b) $X_s\left(e^{j\omega_1}, e^{j\omega_2}\right)$

(c) $X_s^h\left(e^{j\omega_1}, e^{j\omega_2}\right)$

(d) $X_s^v\left(e^{j\omega_1}, e^{j\omega_2}\right)$

(e) $X_s^d\left(e^{j\omega_1}, e^{j\omega_2}\right)$

Figure 15: Frequency-domain diagrams showing a rectangular sampling example (hashed areas are negative, solid areas are positive).

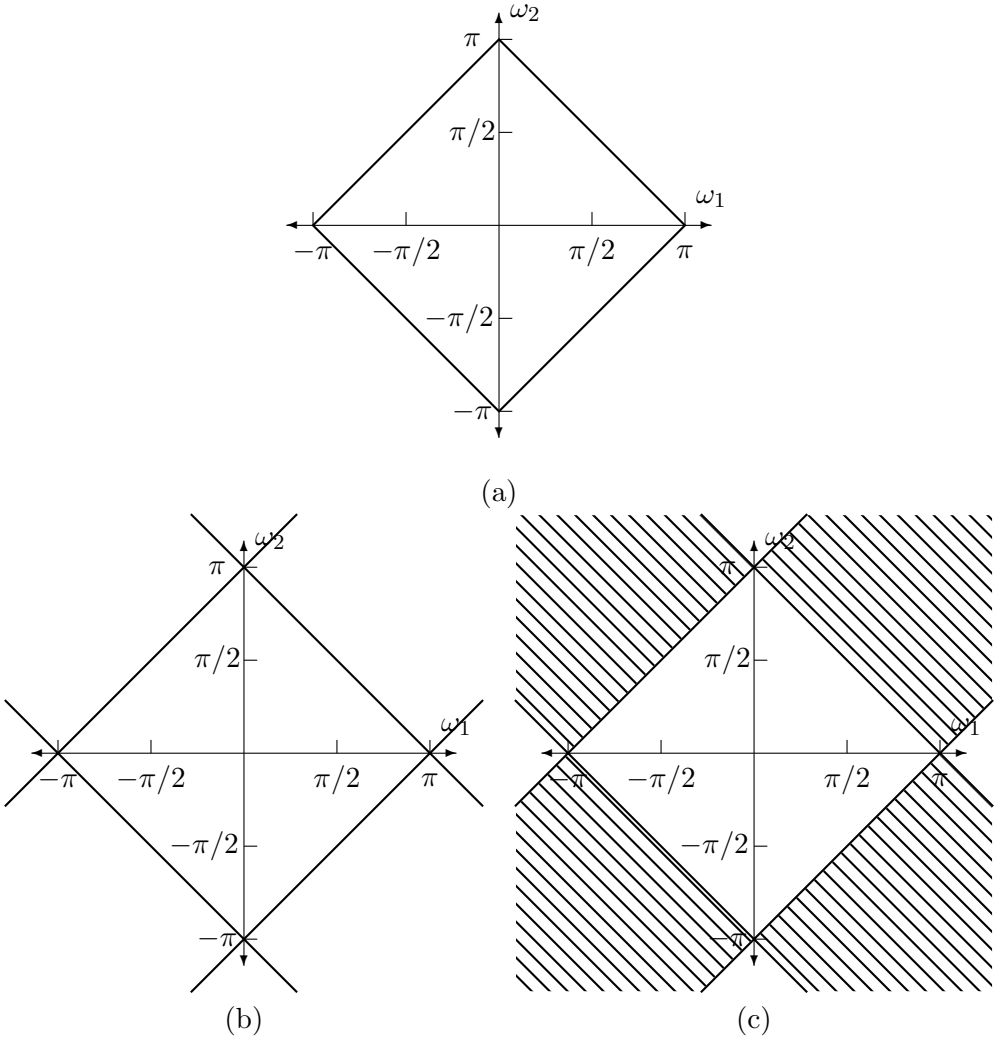


Figure 16: Frequency-domain diagrams showing a quincunx sampling example (hashed areas are negative, solid areas are positive).

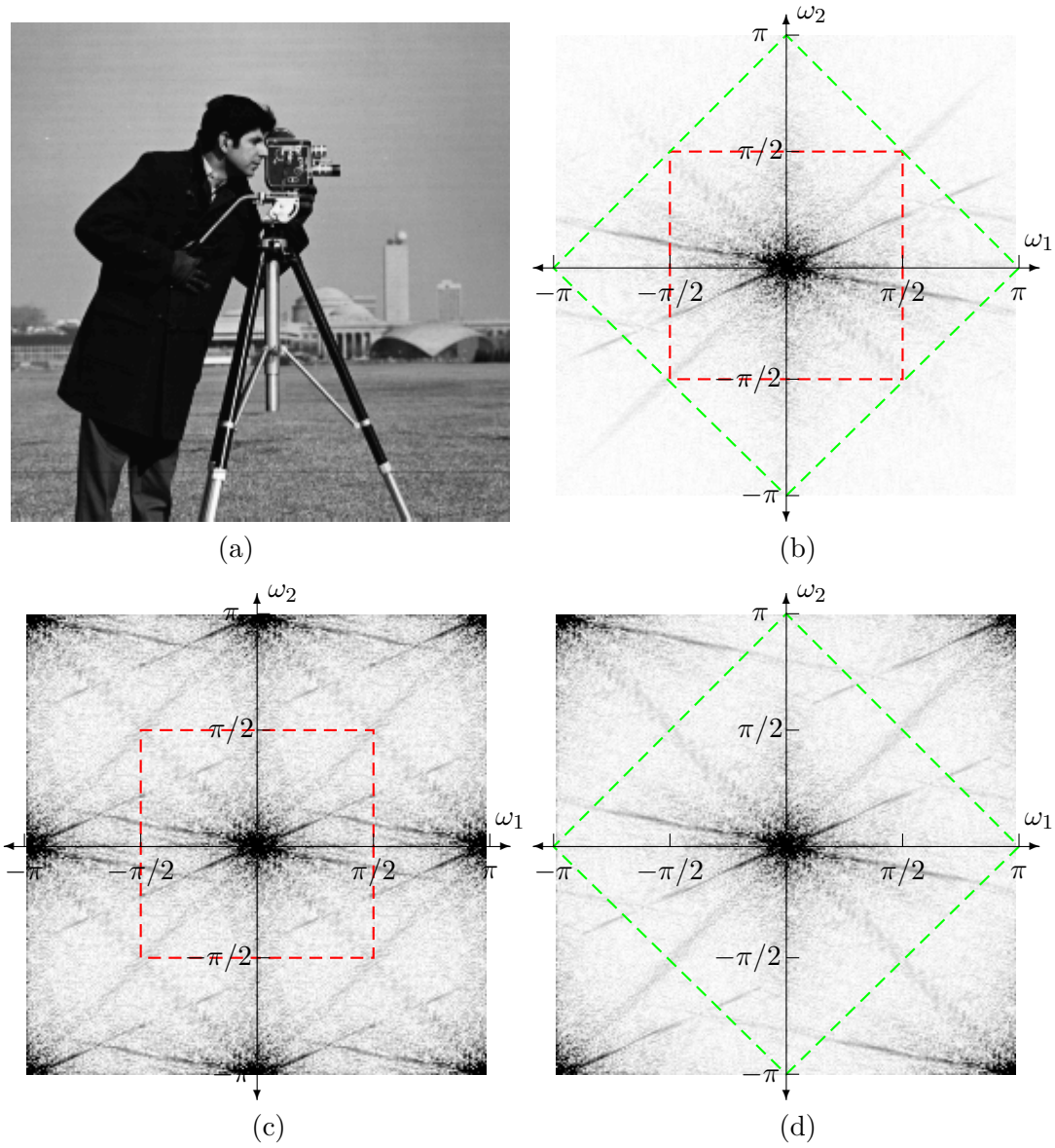


Figure 17: *Cameraman* is used to show the replication of the original spectrum when sampled with rectangular and quincunx grids. (a) Original image. (b) Original spectrum. (c) Sampled with a rectangular grid. (d) Sampled with a quincunx grid.

CHAPTER III

EDGE-DIRECTED INTERPOLATION

Two algorithms from the literature are helpful in analyzing the CFA interpolation problem, the Adams and Hamilton (AH) algorithm and the Projections onto Convex Sets (POCS) algorithm. When interpolating the green image, both use an edge-directed interpolation scheme to determine the best interpolation direction (horizontal or vertical) for each pixel. For the red and blue image interpolation, the AH algorithm uses a simple method that generates good results while the POCS algorithm uses an iterative scheme. This chapter analyzes both of these algorithms in order to gain a better understanding of the CFA interpolation problem from a standard signal processing point of view.

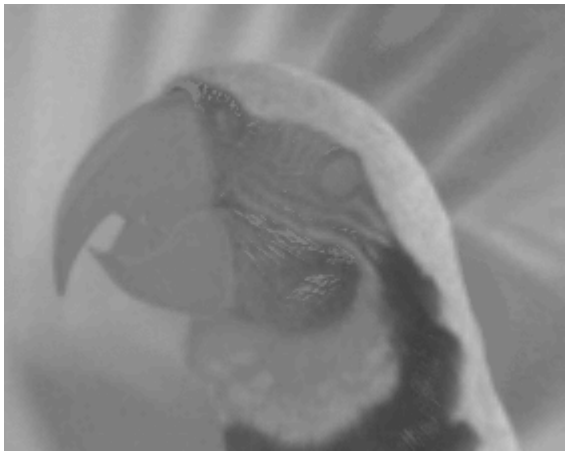
POCS and AH make a common assumption that the red, green, and blue images are perfectly correlated. This assumption is described by the following equations,

$$G[n_1, n_2] = R[n_1, n_2] + k_r \tag{23}$$

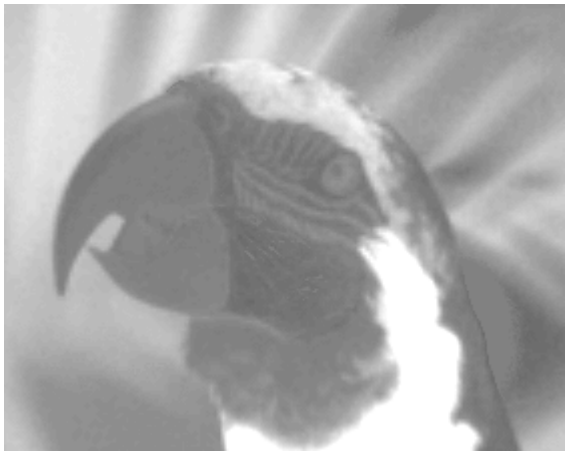
$$G[n_1, n_2] = B[n_1, n_2] + k_b, \tag{24}$$

where k_r and k_b are constants over a small neighborhood in the image. This assumption describes a set of images that have identical high-frequency content and only differ in their low-frequency content. Adams justifies this assumption with an example image [1], shown in Figure 18. In this example, the green image is a rapidly changing image, yet the difference images, $R - G$ and $B - G$, are both fairly smooth. Gunturk et al. report very high correlation values between the red, green, and blue images [12]. This assumption justifies the use of cross-channel interpolation instead of interpolating each channel independently. This chapter describes how the red, green, and blue information can be used to successfully interpolate the CFA image.

(a)



(b)



(c)

Figure 18: *Parrot* image example justifies perfect correlation assumption. (a) Original parrot image. (b) $R - G$ image. (c) $B - G$ image.

3.1 Adams and Hamilton Algorithm

Adams and Hamilton published an edge-directed interpolation algorithm in a series of papers and patents [13, 3, 4, 1, 2]. This algorithm determines whether each pixel belongs to a horizontal feature or a vertical feature and interpolates appropriately for each case. This is done by analyzing the data around each pixel to determine if a horizontal or vertical edge is present. Figure 19 shows two regions of the *Lighthouse* image. The building section has primarily horizontal features and the fence area has primarily vertical features. The edge-directed interpolation algorithm performs better than the standard interpolation algorithms in these areas.

(a) (b)

Figure 19: Edge directed interpolation applied to *Lighthouse* image. (a) Horizontal features. (b) Vertical features.

The mechanics of the algorithm are introduced followed by an analysis and discussion of the algorithm.

3.1.1 Algorithm Definition

The algorithm begins by interpolating the green image using edge-directed interpolation. Each missing green pixel lies in a region like the one in Figure 20. The red pixels are

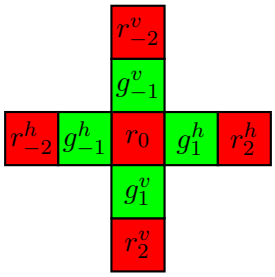


Figure 20: Interpolation region for missing green pixel.

interchanged with blue pixels when the missing green pixel is on a blue pixel. Two classifiers

are computed for the horizontal and vertical directions for each missing green pixel.

$$C_h = 2\left|g^h_{-1} - g^h_1\right| + \left|2r_0 - r^h_{-2} - r^h_2\right| \tag{25}$$

$$C_v = 2\left|g^v_{-1} - g^v_1\right| + \left|2r_0 - r^v_{-2} - r^v_2\right| \tag{26}$$

The green image is being interpolated, but the red and blue images also contain information about edges in the image. Both green and red/blue information are used to determine the interpolation direction. The direction with the smaller classifier is chosen as the interpolation direction. The green value is then estimated from the red and green values along the row or column of the missing pixel, depending on the interpolation direction. If $C_h < C_v$, then the missing green value, $\hat{g}$, is computed using

$$\hat{g} = \frac{1}{2}\left(g^h_{-1} + g^h_1\right) + \left(r_0 - \frac{1}{2}r^h_2 - \frac{1}{2}r^h_{-2}\right). \tag{27}$$

Otherwise, $\hat{g}$ is computed with

$$\hat{g} = \frac{1}{2}\left(g^v_{-1} + g^v_1\right) + \left(r_0 - \frac{1}{2}r^v_2 - \frac{1}{2}r^v_{-2}\right). \tag{28}$$

The interpolation equations show that the missing green value is found by averaging the adjacent green pixels and then adding a correction term from the red or blue pixels in the row or column of the missing pixel.

The red and blue images are interpolated without using a directional decision. The interpolation is separated into three cases. The first case is when the missing red pixel is on a green pixel in a row of green and red pixels. The second case is on a green pixel in a row of green and blue pixels. Here, the adjacent red pixels are aligned vertically. The third case is on a blue pixel, where four red pixels are diagonal from the missing red pixel. These three cases are shown in Figure 21.

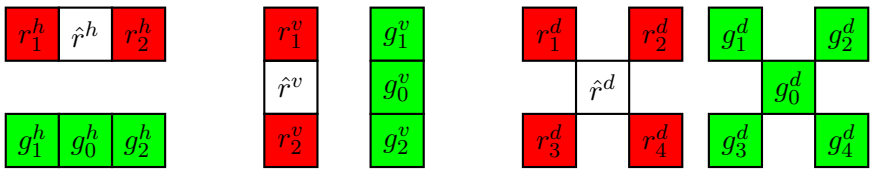


Figure 21: Three cases for red (and blue) interpolation.

Because the green image has been interpolated completely before interpolating the red and blue images, each pixel has a green value. These green values are used for correction terms when interpolating the red and blue images. For the case in Figure 21(a), the red pixels are horizontally adjacent to the missing pixel. The red value is computed using

$$\hat{r}^h = \frac{1}{2}\left(r_1^h + r_2^h\right) + \left(g_0^h - \frac{1}{2}g_1^h - \frac{1}{2}g_2^h\right). \tag{29}$$

The vertical case is shown in Figure 21(b) and the missing value is found by evaluating the same equation, except along the column,

$$\hat{r}^v = \frac{1}{2}\left(r_1^v + r_2^v\right) + \left(g_0^v - \frac{1}{2}g_1^v - \frac{1}{2}g_2^v\right). \tag{30}$$

In the diagonal case, shown in Figure 21(c), the missing red value has four red neighbors at the corners of the 3x3 neighborhood.

$$\hat{r}^d = \frac{1}{4}\left(r_1^d + r_2^d + r_3^d + r_4^d\right) + \left(g_0^d - \frac{1}{4}g_1^d - \frac{1}{4}g_2^d - \frac{1}{4}g_3^d - \frac{1}{4}g_4^d\right) \tag{31}$$

Note that ignoring the green terms in (29), (30), and (31), these equations match those of bilinear interpolation, where the nearest neighbors are averaged to compute the missing value.

The remainder of this section analyzes the AH algorithm to show how this method gives good interpolation results.

3.1.2 Interpolating the Green Image

The green image is interpolated by choosing an interpolation direction, then averaging the neighboring green pixels along that direction. A correction term from the red or blue pixels along that direction is added to the average. If the interpolation direction has been chosen, this system can be represented by a linear system, shown in Figure 22. The green samples, G_s, are filtered by H_1 and the red (or blue) samples, R_s, are filtered by H_2.

We will think about this interpolation in two steps. First, the green samples are used to interpolate the missing green values. A correction term from the red and blue images is added to this initial estimate. Figure 23 shows the green image of a small region of *Lighthouse*. The first image shows only the estimate using the green samples. The second

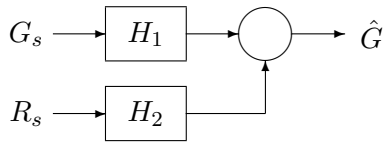


Figure 22: A simplified form of the Adams and Hamilton edge-directed interpolation system for the green image.

(a) (b)

Figure 23: Adding correction terms from the red and blue images improves the green interpolation. (a) Interpolated green image using only green samples. (b) Interpolated green image with correction terms included.

image shows the improvement from adding the correction terms. Aliasing artifacts along the rails of the lighthouse are eliminated in the improved image.

Consider interpolating the green values in a single row when the interpolation direction is horizontal. This example reduces the system to a one-dimensional problem. The input to the system is an interleaved set of samples from two input signals, the green samples and the red samples, whose Fourier transforms are

$$G_s(e^{j\omega}) = \frac{1}{2}G(e^{j\omega}) + \frac{1}{2}G(e^{j(\omega-\pi)}) \tag{32}$$

$$R_s(e^{j\omega}) = \frac{1}{2}R(e^{j\omega}) - \frac{1}{2}R(e^{j(\omega-\pi)}). \tag{33}$$

Therefore, the output equation is given by

$$\begin{aligned}\hat{G}(e^{j\omega}) &= \frac{1}{2}G(e^{j\omega})H_1(e^{j\omega}) + \frac{1}{2}R(e^{j\omega})H_2(e^{j\omega}) + \\ &\quad \frac{1}{2}G(e^{j(\omega-\pi)})H_1(e^{j\omega}) - \frac{1}{2}R(e^{j(\omega-\pi)})H_2(e^{j\omega})\end{aligned} \tag{34}$$

where

$$H_1(e^{j\omega}) = 1 + \cos(\omega)$$

and

$$H_2(e^{j\omega}) = 1 - \cos(2\omega).$$

The filter magnitude responses are shown in Figure 24. The green filter, H_1, is a lowpass filter, but the passband roll-off is severe and the stopband attenuation is small. These limitations suppress medium frequencies in the output image and allow aliasing distortions from the third term in (34) into the output image.

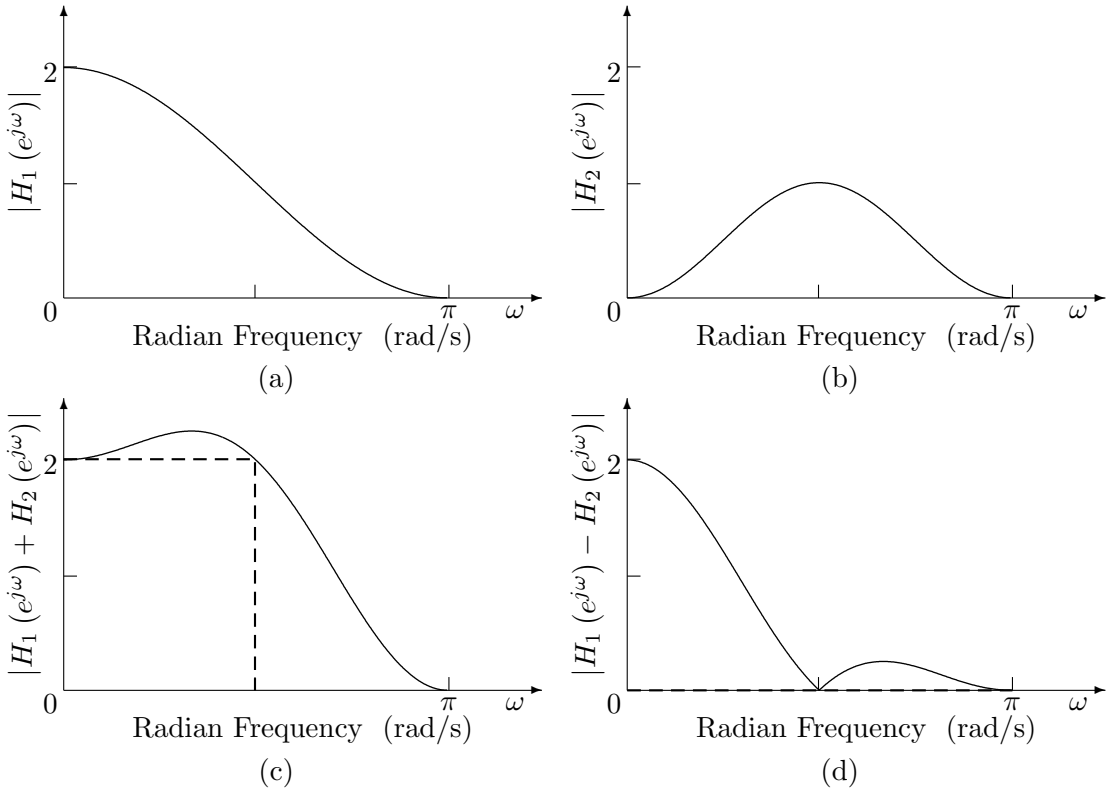


Figure 24: Magnitude response for filters used in AH. (a) H_1. (b) H_2. (c) $H_1 + H_2$ (ideal - dashed). (d) $H_1 - H_2$ (ideal - dashed).

The red terms are used to correct these problems. We make the assumption that the red and green images are perfectly correlated. This assumption says that the red and green images differ only by a constant offset,

$$g[n] = r[n] + k. \tag{35}$$

In the frequency domain, this model is transformed to

$$G(e^{j\omega}) = R(e^{j\omega}) + k \cdot \delta(\omega) \tag{36}$$

where $\delta(\omega)$ is the unit impulse function. Therefore, the spectrum of the green image is identical to the spectrum of the red image, except at $\omega = 0$. Since H_2 is zero at $\omega = 0$ and $G(e^{j\omega}) = R(e^{j\omega})$ at all points except $\omega = 0$, the first two terms of (34) describe the attenuation of the original, non-shifted copy of the spectrum. The response of the sum filter, $H_1 + H_2$, is shown in Figure 24(c). This curve shows that the passband response does not attenuate the medium frequencies as much as the standard filter, H_1.

The third and fourth terms of (34) are the aliasing distortion terms. Therefore, we can analyze the performance of the system with respect to aliasing by examining the filter response of the difference filter, $H_1 - H_2$. This is shown in Figure 24(d). Between $\pi/2$ and π, the difference filter is nearly zero. Therefore, in the region where most of the energy in the shifted spectrum exists, the system has high attenuation, thus reducing the amount of aliasing in the output image.

A two-dimensional chirp signal is a useful artificial signal to illustrate the response of a system. The chirp signal sweeps the frequency space,

$$\{(\omega_1, \omega_2) : -\pi \leq \omega_1 < \pi, \quad -\pi \leq \omega_2 < \pi\},$$

such that the instantaneous frequency of the signal is equal to the coordinates in the frequency space. This allows us to visualize how the system responds to all frequencies. The chirp image is shown in Figure 25(a). An example of how this signal can be used to show the system response is given in this figure. A one-dimensional lowpass filter and a highpass filter are designed. The responses of these filters are shown in Figure 25(b). These filters are applied to the rows of the synthetic chirp image. Figure 25(c) shows the output when the lowpass filter is applied and Figure 25(d) shows the output when the highpass filter is applied. In the lowpass image, only the central part of the image, where the horizontal frequency is small, is not blocked by the filter. In the highpass image, the regions of high horizontal frequency along the left and right edges of the image are not blocked while the central portion of the image is blocked. Thus, the frequency response of a system can be seen by using this chirp image as the input to the system.

Figure 26 shows the system response of the AH algorithm using a two-dimensional chirp

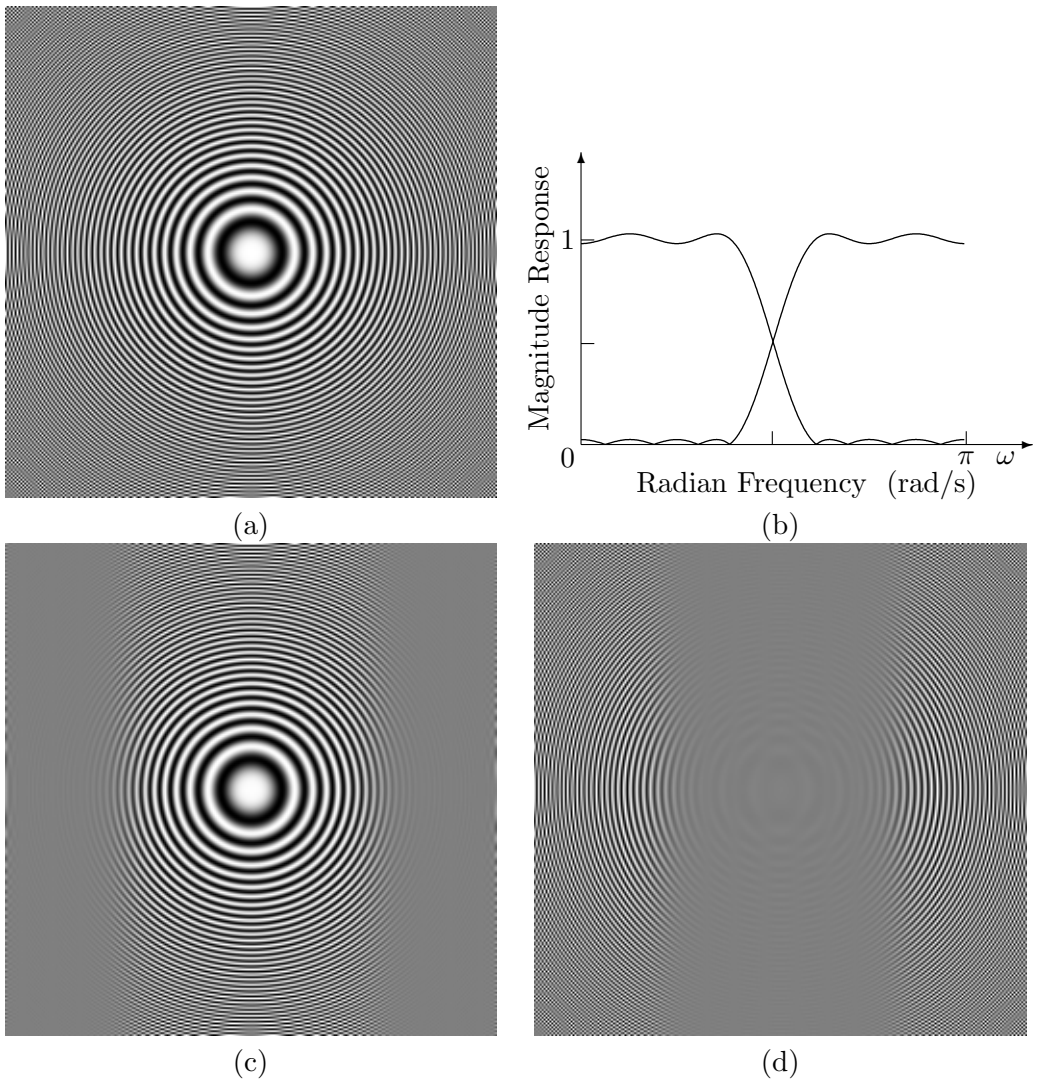
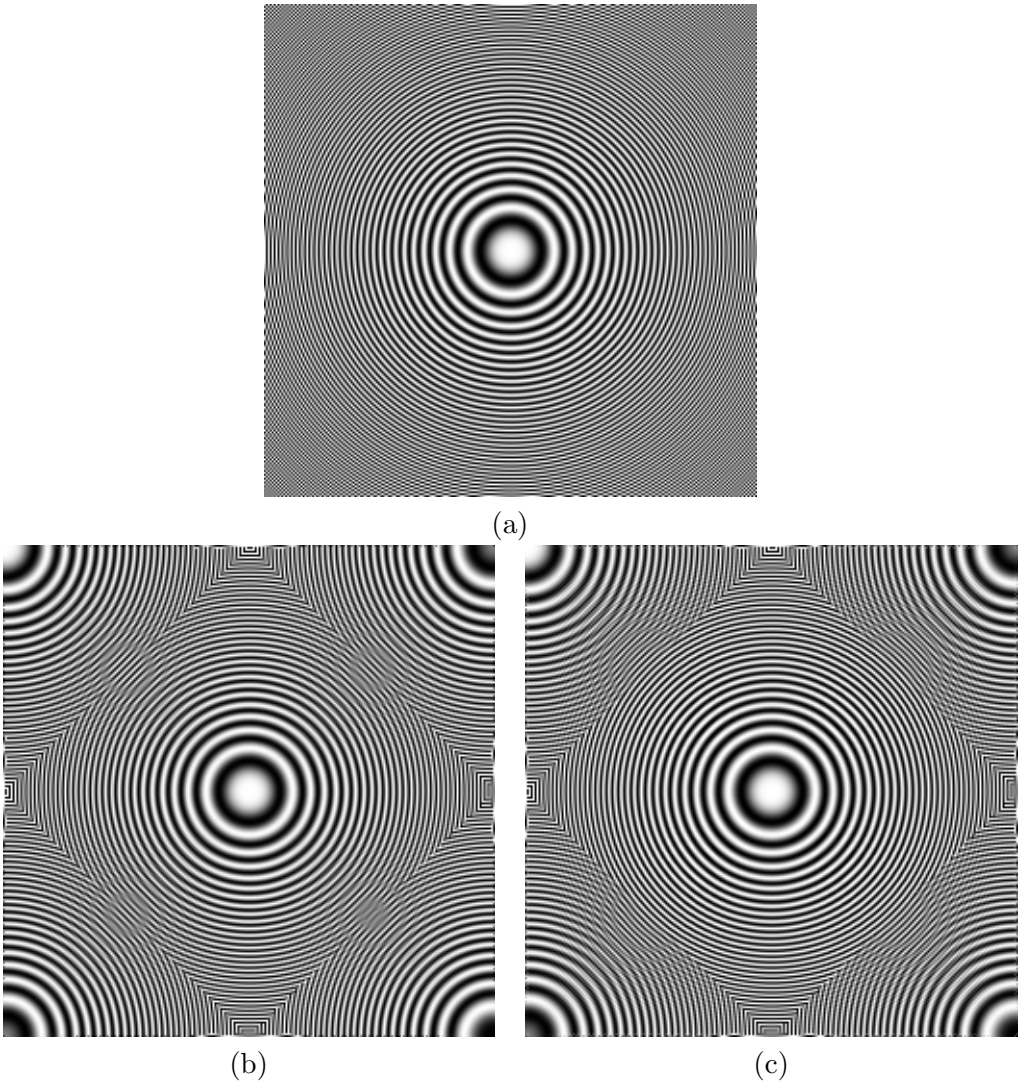


Figure 25: The chirp image can be used to illustrate the frequency response of a system. (a) Synthetic chirp image. (b) Lowpass and highpass filters used in example. (c) Lowpass image. (d) Highpass image.

signal. The original image, shown in Figure 26(a), is a gray image, so the red image, green image, and blue image are identical. This is the ideal case ensuring that the perfect correlation assumption is met. Figure 26(b) shows the interpolated green image without using the correction terms from the red and blue images. This image shows the diamond-shaped Nyquist region from the quincunx sampling grid for the green image. Along the edges of the diamond, however, aliasing rings can be seen. These are distortions that are not seen in the original image. Without the correction, this edge-directed system has problems reconstructing high-frequency components with a diagonal orientation. To obtain Figure

26(c), the red and blue correction terms are added to the interpolated green image. The aliasing rings in the previous image are not present in the corrected image. In fact, the image shows how, outside of the diamond-shaped region, the aliasing rings are present, but they are corrected inside the diamond-shaped region. This image verifies that this algorithm can better reconstruct all frequencies within the diamond-shaped region. Even though the algorithm chooses between a horizontal and vertical orientation, it better reconstructs diagonal components by adding the red and blue correction terms.



(a)

(b) (c)

Figure 26: Response of AH to two-dimensional chirp signal. The chirp signal sweeps the frequency space $[-\pi, \pi] \times [-\pi, \pi]$. The system response is visualized by the output image. (a) Original chirp image. (b) Reconstructed image using edge-directed interpolation without adding correction terms from the red and blue images. (c) Reconstructed image adding correction terms to result in (b).

3.1.3 Interpolating the Red and Blue Images

As described in Section 3.1.1, AH interpolates the red and blue images for three different cases. Figure 21 is repeated here as Figure 27 to show these cases. Ignoring the green terms, the red interpolation begins with the following equations:

$$\hat{r}^h = \frac{1}{2}\left(r_1^h + r_2^h\right) \tag{37}$$

$$\hat{r}^v = \frac{1}{2}\left(r_1^v + r_2^v\right) \tag{38}$$

$$\hat{r}^d = \frac{1}{4}\left(r_1^d + r_2^d + r_3^d + r_4^d\right). \tag{39}$$

This is equivalent to filtering the red samples with the filter kernel,

$$H_r = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}. \tag{40}$$

This is the same filter kernel used for bilinear interpolation. Therefore, this algorithm begins by using bilinear interpolation to generate an initial estimate of the red and blue images. As noted in Section 1.1, this does not provide good results. But the result image in Figure 6 is a very good result. Therefore, much improvement is made by the green correction terms.

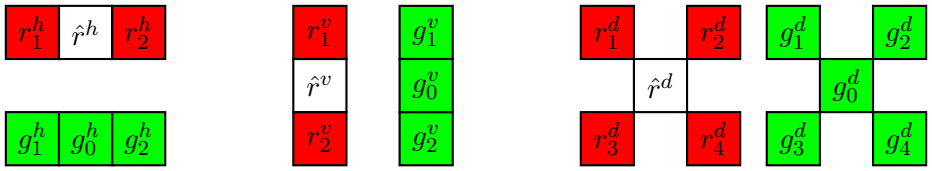


Figure 27: Three cases for red (and blue) interpolation.

The correction signal can be derived by starting with the correction signals for each of the three cases,

$$c_g^h = \left(g_0^h - \frac{1}{2}g_1^h - \frac{1}{2}g_2^h\right) \tag{41}$$

$$c_g^v = \left(g_0^v - \frac{1}{2}g_1^v - \frac{1}{2}g_2^v\right) \tag{42}$$

$$c_g^d = \left(g_0^d - \frac{1}{4}g_1^d - \frac{1}{4}g_2^d - \frac{1}{4}g_3^d - \frac{1}{4}g_4^d\right). \tag{43}$$

For each of these three cases, we can define a filter kernel that is applied to the interpolated green image to generate the correction signal,

$$H_h = \begin{bmatrix} -1/2 & 1 & -1/2 \end{bmatrix} \quad H_v = \begin{bmatrix} -1/2 \\ 1 \\ -1/2 \end{bmatrix} \quad H_d = \begin{bmatrix} -1/4 & 0 & -1/4 \\ 0 & 1 & 0 \\ -1/4 & 0 & -1/4 \end{bmatrix}. \tag{44}$$

In the frequency domain, the initial estimate of the interpolated image is found with

$$\hat{R}\left(e^{j\omega_1}, e^{j\omega_2}\right) = R_s\left(e^{j\omega_1}, e^{j\omega_2}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right). \tag{45}$$

Substituting for R_s the sampling equation in (20) gives

$$\begin{aligned} \hat{R}\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}\Big[R\left(e^{j\omega_1}, e^{j\omega_2}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) \\ &\quad -R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) \\ &\quad +R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) \\ &\quad -R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right)\Big]. \end{aligned} \tag{46}$$

Determining the correction signal is more difficult since three different filters are used – one for each of the three cases. For green pixels in a red row (Gr pixels), the correction signal is

$$C_h\left(e^{j\omega_1}, e^{j\omega_2}\right) = \hat{G}\left(e^{j\omega_1}, e^{j\omega_2}\right) H_h\left(e^{j\omega_1}, e^{j\omega_2}\right). \tag{47}$$

The correction signal for green pixels in a blue row (Gb pixels) is

$$C_v\left(e^{j\omega_1}, e^{j\omega_2}\right) = \hat{G}\left(e^{j\omega_1}, e^{j\omega_2}\right) H_v\left(e^{j\omega_1}, e^{j\omega_2}\right). \tag{48}$$

For blue pixels (B pixels), the correction signal is

$$C_d\left(e^{j\omega_1}, e^{j\omega_2}\right) = \hat{G}\left(e^{j\omega_1}, e^{j\omega_2}\right) H_d\left(e^{j\omega_1}, e^{j\omega_2}\right). \tag{49}$$

At red pixels (R pixels), there is no correction; the sampled value is passed to the output unchanged. The correction signal is derived by sampling the outputs of the correction filters at the appropriate outputs, as shown in Figure 28. The Gr pixels correspond to the sampling

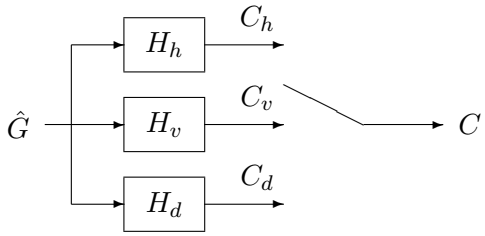


Figure 28: System used by AH to correct the red and blue images using the green image.

grid in Figure 9(a), whose sampling equation is given by (5). Therefore, the correction term is given by

$$\begin{aligned}\tilde{C_h}\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}\left[\hat{G}\left(e^{j\omega_1},e^{j\omega_2}\right)H_h\left(e^{j\omega_1},e^{j\omega_2}\right)\right.\\ &+\hat{G}\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)H_h\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)\\ &+\hat{G}\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)H_h\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)\\ &\left.+\hat{G}\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)H_h\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)\right]. \qquad (50)\end{aligned}$$

The Gb pixels use the sampling grid in Figure 9(d) and the B pixels use the sampling grid in Figure 9(c). The sampling equation for the Gb pixels is given by (11) and the equation for the B pixels is given by (10). The correction terms for the Gb and B pixels are given by

$$\begin{aligned}\tilde{C_v}\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}\left[\hat{G}\left(e^{j\omega_1},e^{j\omega_2}\right)H_v\left(e^{j\omega_1},e^{j\omega_2}\right)\right.\\ &-\hat{G}\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)H_v\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)\\ &-\hat{G}\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)H_v\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)\\ &\left.+\hat{G}\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)H_v\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)\right] \qquad (51)\end{aligned}$$

and

$$\begin{aligned}\tilde{C_d}\left(e^{j\omega_1},e^{j\omega_2}\right) &= \frac{1}{4}\left[\hat{G}\left(e^{j\omega_1},e^{j\omega_2}\right)H_d\left(e^{j\omega_1},e^{j\omega_2}\right)\right.\\ &+\hat{G}\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)H_d\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)\\ &-\hat{G}\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)H_d\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)\\ &\left.-\hat{G}\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)H_d\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)\right]. \qquad (52)\end{aligned}$$

The correction signal is found by combining (50), (51), and (52) and collecting terms, as in

$$C\left(e^{j\omega_1},e^{j\omega_2}\right) =$$

$$
\begin{aligned}
&\frac{1}{4}\left\{\hat{G}\left(e^{j\omega_1}, e^{j\omega_2}\right)\cdot\left[H_h\left(e^{j\omega_1}, e^{j\omega_2}\right)+H_v\left(e^{j\omega_1}, e^{j\omega_2}\right)+H_d\left(e^{j\omega_1}, e^{j\omega_2}\right)\right]\right.\\
&+\hat{G}\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)\cdot\left[H_h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)-H_v\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)+H_d\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)\right]\\
&+\hat{G}\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)\cdot\left[H_h\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)-H_v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)-H_d\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)\right]\\
&+\hat{G}\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\cdot\\
&\left.\left[H_h\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)+H_v\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)-H_d\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right]\right\}. \qquad (53)
\end{aligned}
$$

Each of the four terms in (53) applies a combination of the three correction filters to a shifted replica of the original spectrum. In the first term, the filter combination is $H_h + H_v + H_d$. Substituting the filter kernels from (44) into this expression gives the effective correction filter,

$$
\begin{aligned}
H_h + H_v + H_d &= \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 1 & -1/2 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1/2 & 0 \\ 0 & 1 & 0 \\ 0 & -1/2 & 0 \end{bmatrix} + \begin{bmatrix} -1/4 & 0 & -1/4 \\ 0 & 1 & 0 \\ -1/4 & 0 & -1/4 \end{bmatrix} \\
&= \begin{bmatrix} -1/4 & -1/2 & -1/4 \\ -1/2 & 3 & -1/2 \\ -1/4 & -1/2 & -1/4 \end{bmatrix}. \qquad (54)
\end{aligned}
$$

Using the same argument from interpolating the green image, we define an effective filter for $R\left(e^{j\omega_1}, e^{j\omega_2}\right)$ by combining the red interpolation filter with the correction filter in (54).

$$
\begin{aligned}
H_r + H_h + H_v + H_d &= \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} + \begin{bmatrix} -1/4 & -1/2 & -1/4 \\ -1/2 & 3 & -1/2 \\ -1/4 & -1/2 & -1/4 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (55)
\end{aligned}
$$

The resulting filter is an identity filter, with a scale factor of 4 to invert the 1/4 scaling of the sampling system. This is an interesting result because it says that the non-shifted term, $R\left(e^{j\omega_1}, e^{j\omega_2}\right)$, will be perfectly reconstructed.

For the other terms, the frequency shift in the correction terms needs to be accounted for before computing an effective filter. Because

$$e^{j\omega_o n}x[n] \longleftrightarrow X\left(e^{j(\omega-\omega_o)}\right),$$

a frequency shift of π rad/s is equivalent to multiplying the spatial signal by $(-1)^n$. For two-dimensional signals, three frequency-shift properties exist:

$$e^{j\omega_o n_1}x[n_1,n_2] \longleftrightarrow X\left(e^{j(\omega_1-\omega_o)},e^{j\omega_2}\right) \tag{56}$$

$$e^{j\omega_o n_2}x[n_1,n_2] \longleftrightarrow X\left(e^{j\omega_1},e^{j(\omega_2-\omega_o)}\right) \tag{57}$$

$$e^{j\omega_o n_1}e^{j\omega_o n_2}x[n_1,n_2] \longleftrightarrow X\left(e^{j(\omega_1-\omega_o)},e^{j(\omega_2-\omega_o)}\right). \tag{58}$$

These properties are applied to the correction terms in (53) to give effective filters for $R\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$, $R\left(e^{j\omega_1},e^{j(\omega_2-\pi)}\right)$, and $R\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)$. To compute the effective filters, the like terms are gathered from (53) and (46), collecting terms as if $R=\hat{G}$. This simplification arises from the perfect correlation assumption where we assume that the red, green, and blue signals only differ in their low-frequency content. Since H_h, H_v, and H_d are highpass filters, low-frequency information from the green image is not mixed with the red image.

The like terms from (53) and (46) for the factors $R\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$ and $\hat{G}\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$ are: $-H_r\left(e^{j\omega_1},e^{j\omega_2}\right)$, $H_h\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$, $-H_v\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$, and $H_d\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$. The filter kernels corresponding to these terms are computed. For $-H_r\left(e^{j\omega_1},e^{j\omega_2}\right)$, the coefficients from H_r in (40) are multiplied by -1. Using the property in (56), the kernels for the terms $H_h\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$, $-H_v\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$, and $H_d\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)$ are found by multiplying H_h, H_v, and H_d in (44) by $(-1)^{n_1}$, marking the center elements of H_h, H_v, and H_d with the coordinate $[0,0]$. This effective filter is computed in (59).

$$-H_r\left(e^{j\omega_1},e^{j\omega_2}\right)+H_h\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)-H_v\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right)+H_d\left(e^{j(\omega_1-\pi)},e^{j\omega_2}\right):$$

$$\begin{bmatrix} -1/4 & -1/2 & -1/4 \\ -1/2 & -1 & -1/2 \\ -1/4 & -1/2 & -1/4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 1 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1/2 & 0 \\ 0 & 1 & 0 \\ 0 & -1/2 & 0 \end{bmatrix} + \begin{bmatrix} 1/4 & 0 & 1/4 \\ 0 & 1 & 0 \\ 1/4 & 0 & 1/4 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{59}$$

The same process is used to compute the effective filter for the $R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$ term. The property in (57) is used to find the kernels for $H_h\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$, $H_v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$, and $H_d\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$. The filter kernels, H_h, H_v, and H_d are multiplied by $(-1)^{n_2}$. This effective filter is computed in (60).

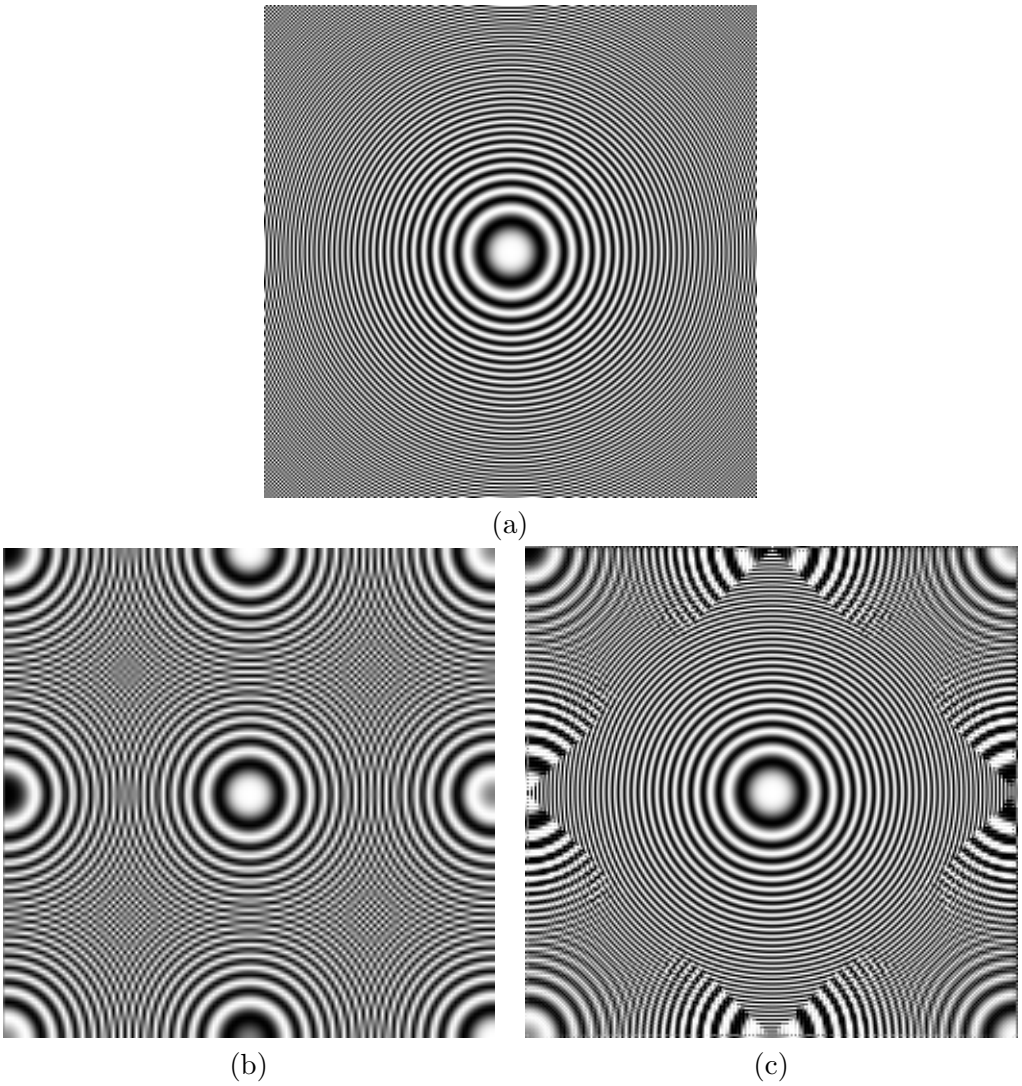
$$\begin{aligned} &H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_h\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - H_v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - H_d\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right): \\ &= \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 1 & -1/2 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} - \begin{bmatrix} 1/4 & 0 & 1/4 \\ 0 & 1 & 0 \\ 1/4 & 0 & 1/4 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{60}$$

The kernels for $H_h\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$, $H_v\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$, and $H_d\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$ are computed using the property in (58). The kernels H_h, H_v, and H_d are multiplied by $(-1)^{n_1+n_2}$. The effective filter for the $R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$ term is computed in (61).

$$\begin{aligned} &-H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_h\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) + H_v\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) - H_d\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) \\ &= \begin{bmatrix} -1/4 & -1/2 & -1/4 \\ -1/2 & -1 & -1/2 \\ -1/4 & -1/2 & -1/4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 1 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} - \begin{bmatrix} -1/4 & 0 & -1/4 \\ 0 & 1 & 0 \\ -1/4 & 0 & -1/4 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{61}$$

All of the effective filters reduce to zero filters so each of the aliasing terms in the sampled signal is cancelled by adding the green correction terms. The assumption of perfect correlation between the red and green images must hold for the analysis of the effective

filters to be meaningful. This result explains how the AH algorithm performs so well. Any aliasing components that pass through the bilinear interpolation step are cancelled when the green correction terms are added to the result. Also, the green correction terms add high-frequency information to the interpolated red image. This adds sharpness to edges in the image.



(a)

(b) (c)

Figure 29: Two-dimensional chirp signal as input to red/blue AH interpolation system. AH removes much of the aliasing present in the bilinear interpolation result. (a) Original chirp image. (b) Reconstructed image using bilinear interpolation. (c) Reconstructed using correction terms from green image.

Figure 29 shows the chirp signal applied to the red interpolation system of the AH

algorithm. The original image is shown in Figure 29(a) and the result using bilinear interpolation is shown in Figure 29(b). This image shows clearly the aliasing that results from high frequencies in the input image. Outside the square Nyquist region, all high-frequency components are reconstructed as low-frequency components. In the analysis of this system, it was shown that the aliasing terms would be completely cancelled. Figure 29(c) shows the image after adding the green correction terms. This image shows dramatically reduced aliasing distortions. The diamond-shaped region where the green image is reconstructed correctly is apparent in this result. Outside of this region, the green image still has aliasing distortions, so the aliasing is not removed in the red and blue images either. In a small region just outside of the diamond-shaped region along a 45° angle from the origin, the red and blue image is free of aliasing distortions that were present in the green image. The reason for this behavior is unclear.

According to this analysis, adding the correction terms into the interpolated red and blue images, a perfect reconstruction should be possible. Figure 29 shows that aliasing distortions outside of the diamond-shaped region are present. The limitation is the reconstruction of the green image. The green image was unable to correctly interpolate the chirp image outside of the diamond region, so the red and blue images, which use the interpolated green image, would also contain distortions for frequencies outside of the diamond region. In Figure 30, an example image is used to highlight this point. Figure 30(b) shows the result of AH using the interpolated green image for the correction terms and Figure 30(c) shows the result using the original green image for the correction terms. What this example shows is that a large improvement is made when the original green image is used for the corrections. The mean square error (MSE) of the reconstructed image improved from 13.82 to 1.90. For this example, the red and green images are highly correlated, so few errors exist in the reconstruction. This example also shows the dependence on the green interpolation.

(a) (b) (c)

Figure 30: The red *Lighthouse* image interpolated using the AH algorithm. The green correction terms are taken from both the interpolated green image and the original green image. (a) Original red image. (b) Reconstructed red image using interpolated green image for correction terms. (c) Reconstructed red image using original green image for correction terms.

3.1.4 Summary

Edge-directed interpolation is an important algorithm for the CFA interpolation problem. Many algorithms are based on this approach. Adams and Hamilton defined one particular algorithm that works very well and is simple to implement. The simple nature of the algorithm allows us to perform an analysis, which shows the importance of using the correlation between images. For both the red/blue interpolation and the green interpolation, the results were improved by adding correction terms from the other images.

For the green interpolation, an interpolation direction is chosen based on local features in the image. The direction can be either horizontal or vertical. The neighboring green

pixels are averaged along this direction and the Laplacian of the red or blue pixels along that direction is added to the average as a correction term. This correction term removes aliasing in the green image, especially where the image contains angled features. This correction term also enhances medium-frequency components, which the averaging filter attenuates.

The red/blue interpolation starts by using bilinear interpolation to estimate the output image. This initial estimate contains aliasing and lacks high-frequency information, severely distorting the output image. A correction term from the green image is added to the original estimate. A different term is added for each type of pixel (Gr, Gb, or B in the case of red interpolation). Working together, these correction terms reduce aliasing and add high-frequency information in the output image. In fact, the analysis showed that for perfect correlation between the green image and the red/blue image, the reconstruction would be perfect, completely cancelling the aliasing terms and forcing the output image to perfectly match the original input image.

There are some limitations to this seemingly perfect interpolation. In practical situations, the correlation between the images will be high, but not perfect. Thus, combining information among the images may add unwanted distortions. The red and blue interpolation is limited by the interpolation of the green image. Although the analysis reports a perfect reconstruction, if the green image is not interpolated perfectly, the imperfections in the green interpolation will translate into imperfections in the red and blue interpolation.

3.2 Projection onto Convex Sets

The Projections onto Convex Sets (POCS) algorithm is a widely referenced algorithm that uses a filterbank structure to interpolate the CFA images [12]. Like AH, it is divided into two parts, the green interpolation and the red/blue interpolation.

3.2.1 POCS - Green Interpolation

POCS initially interpolates the green image with an edge-directed scheme and then corrects this with information from the red and blue images. The edge-directed portion of the POCS algorithm is identical to that of the AH algorithm, but doesn't add the red and blue

correction terms in the directional interpolation. Instead, the green interpolated values are passed through an analysis filterbank. The green values at red pixels are passed through one filterbank and the red samples are passed through an identical filterbank. The low-frequency subband of the green image is combined with the high-frequency subbands of the red image in a synthesis filterbank that forms an analysis-synthesis perfect reconstruction pair with the analysis filterbank. The output of this filterbank replaces the interpolated values at the red pixels. The same correction is performed at the blue pixels. A block diagram for the green interpolation is presented in Figure 31.

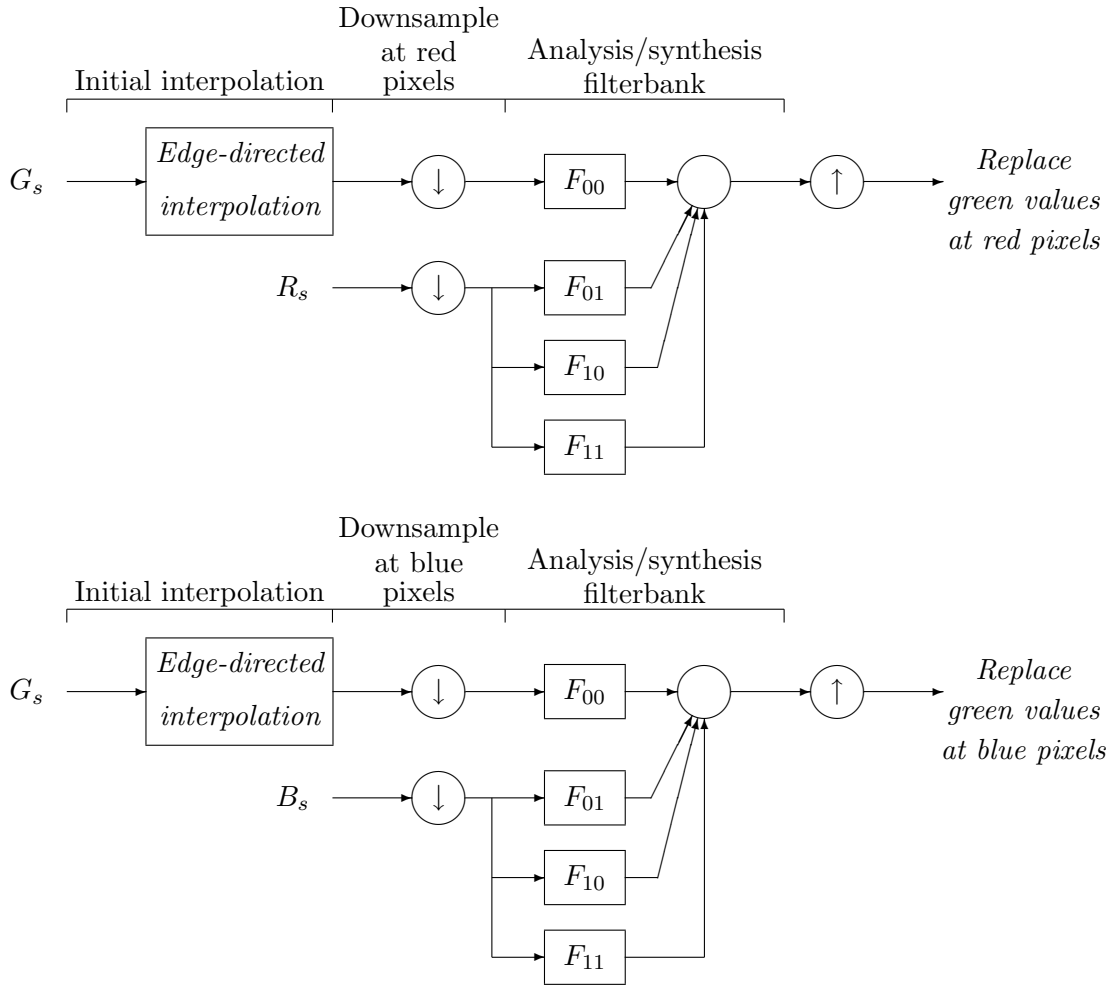


Figure 31: To interpolate the green image, POCS uses edge-directed interpolation, then corrects this using an analysis-synthesis filterbank structure.

As shown in Figure 32, this block diagram is simplified by combining the highpass

filters into a single filter, F_1. We also simplify the green branch by noting that the two

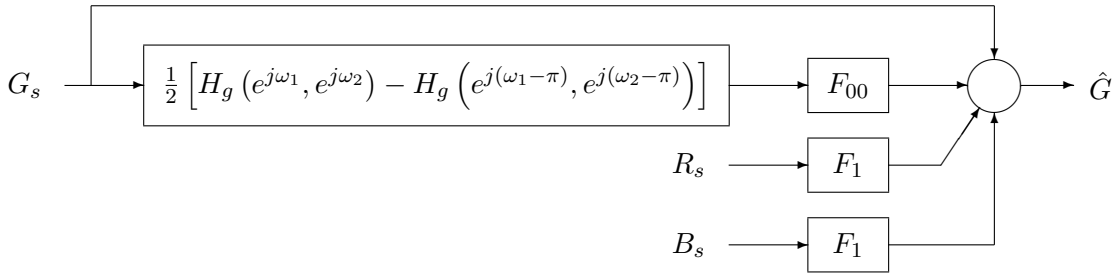


Figure 32: A simplified block diagram for the POCS algorithm. The filter, F_1, is a combined version of the analysis-synthesis filterbank filters for the high-frequency subbands.

paths for the red and blue pixels are identical. Therefore, the path can be rewritten as the green samples filtered by the edge-directed interpolation, keeping only those values at red and blue pixels, and passed through the filterbank. If we describe the edge-directed interpolation filter as a linear filter (assuming that an interpolation direction has been chosen), H_g, then keeping only the values at red and blue pixels would be the same as filtering by $H_g\left(e^{j\omega_1}, e^{j\omega_2}\right) - H_g\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$. This replaces all even coefficients of the interpolation filter, H_g, with zeros so that the output of this filter is non-zero only at red and blue pixels. In the original block diagram, the filterbank filters were applied after a downsampler. In the simplified block diagram, the downsampler and the upsampler are removed; thus, the filters used should be frequency scaled, as in $F\left(e^{j2\omega_1}, e^{j2\omega_2}\right)$.

The filterbank filters combine to give F_{00} and F_1. For completeness, the entire filterbank structure is given in Figure 33. In practice, this complete structure is used, but for this analysis, the branches are combined to give the simplified structure. An example set of perfect reconstruction filters is

$$\begin{aligned} h_0 &= \begin{bmatrix} 0.25 & 0.5 & 0.25 \end{bmatrix} \\ h_1 &= \begin{bmatrix} 0.25 & -0.5 & 0.25 \end{bmatrix} \\ g_0 &= \begin{bmatrix} -0.125 & 0.25 & 0.75 & 0.25 & -0.125 \end{bmatrix} \\ g_1 &= \begin{bmatrix} 0.125 & 0.25 & -0.75 & 0.25 & 0.125 \end{bmatrix}. \end{aligned} \tag{62}$$

From the block diagram in Figure 32, the mathematical description for this system is

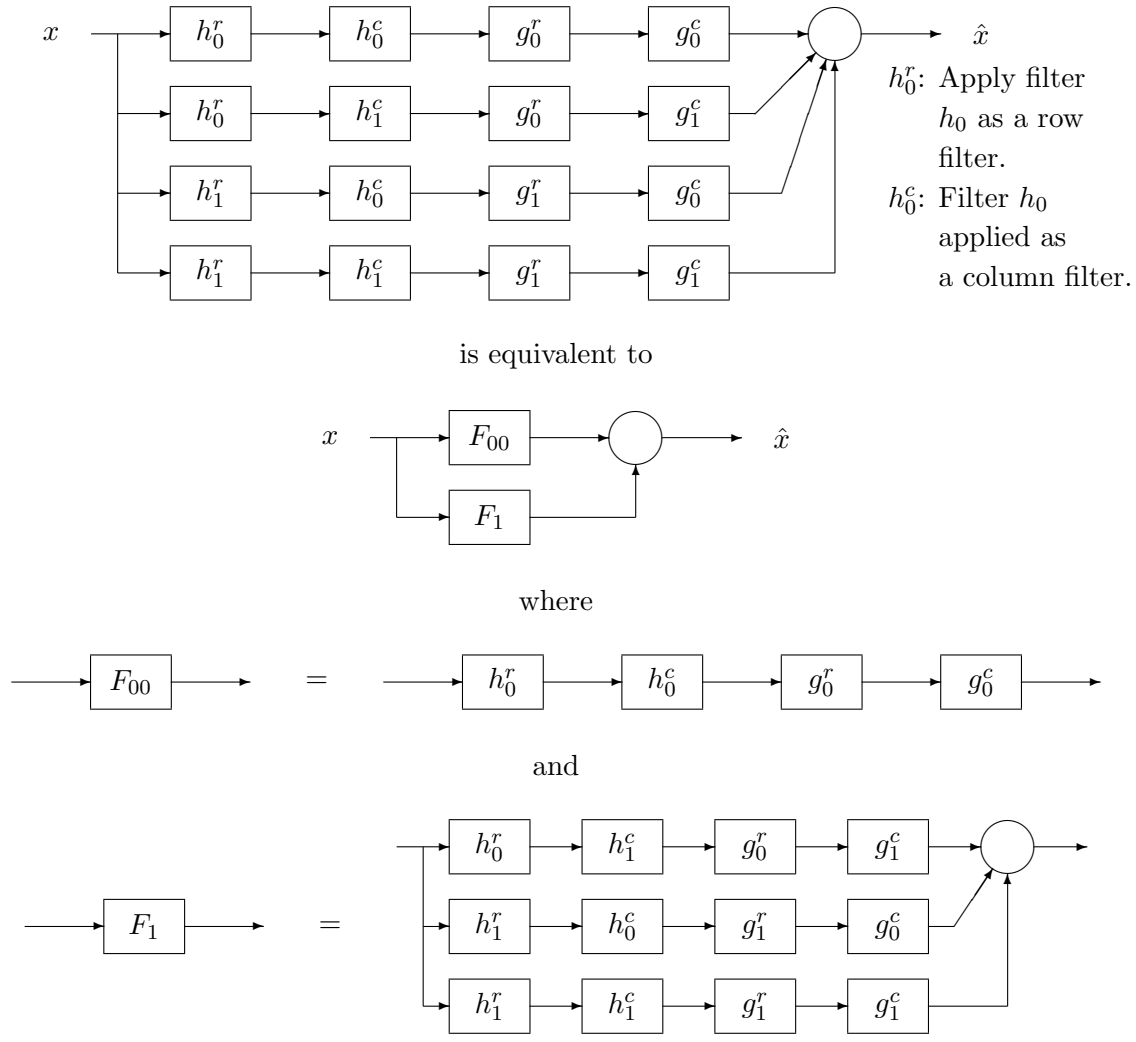


Figure 33: The filterbank structure used in the POCS algorithm.

written as

$$\begin{aligned}\hat{G} &= \frac{1}{2}\left[G\left(e^{j\omega_1}, e^{j\omega_2}\right) + G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right] \cdot \\ &\quad \left[1 + \frac{1}{2}\left(H_g\left(e^{j\omega_1}, e^{j\omega_2}\right) - H_g\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right) F_{00}\left(e^{j2\omega_1}, e^{j2\omega_2}\right)\right] + \\ &\quad \frac{1}{4}\left[R\left(e^{j\omega_1}, e^{j\omega_2}\right) - R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) + \right. \\ &\quad \left. B\left(e^{j\omega_1}, e^{j\omega_2}\right) + B\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - B\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - B\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right] \cdot \\ &\quad F_1\left(e^{j2\omega_1}, e^{j2\omega_2}\right). \end{aligned} \tag{63}$$

In Figure 34, the filter responses of the filters used in the POCS algorithm are shown. In (a), the response of the effective green filter is shown. This filter is the combination of H_g

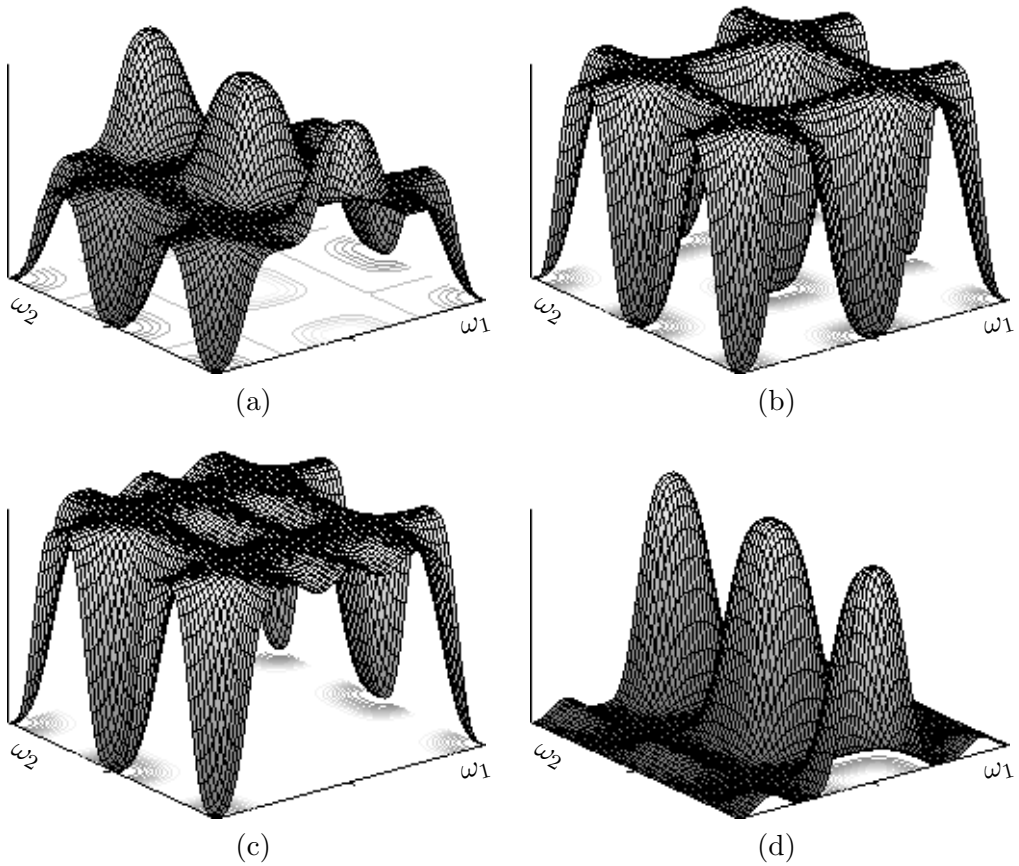


Figure 34: Filter responses for the POCS filters. (a) Filter applied to the green samples. (b) Filter applied to the red and blue samples. (c) Sum of (a) and (b) shows a flat response along the vertical axis. (d) Difference of (a) and (b) shows good aliasing rejection of vertical high frequencies.

and F_{00}, where H_g is assumed to be a horizontal filter, i.e. the horizontal direction has been chosen as the interpolation direction. If the horizontal direction is chosen, then the algorithm has detected more high-frequency information in the vertical direction than in the horizontal direction. This filter blocks the high frequencies in the horizontal direction and passes the high vertical frequencies and the low frequencies. The medium frequencies are attenuated but not blocked. This attenuation is compensated by the filter applied to the red and blue samples, whose response is shown in (b). (63) shows that the $(e^{j\omega_1}, e^{j\omega_2})$ terms from the red, green, and blue images combine. The combination filter for these terms is the sum of the filters in (a) and (b), whose response is shown in (c). Here, it is shown that the response is nearly flat along the vertical direction, where the edge-directed

algorithm has determined that high frequencies exist that should be passed to the output. Along the horizontal direction, high frequencies are blocked. The combination filter for the $\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$ terms is the difference between (a) and (b), shown in (d). This response shows how the system handles aliasing. Noting that this filter is applied to the shifted version of the spectrum, the peaks of the filter align with high horizontal frequencies and high diagonal frequencies. This aliasing will pass to the output if it is present, but the edge detection is using a horizontal filter, so in this case, it is less likely that the image contains horizontal high frequencies. The rest of the spectrum is close to zero, which says that most aliasing due to vertical high frequencies is blocked by this system. The other terms in (63), the $\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)$ terms and the $\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$ terms cancel as long as the red and blue sampled images have the same content in the passband of the F_1 filter. Since this filter blocks frequencies at the origin and the locations of the spectral copies, our assumption that the three images contain similar high-frequency information says that these terms will cancel.

This shows that POCS is a system that relies upon the edge-directed interpolation and the combination of the red, green, and blue information to interpolate the green image. By performing edge detection, aliasing can be nearly completely removed in one direction at the risk of including aliasing from the other direction. Problems would arise if a portion of an image contained both horizontal and vertical features.

Figure 35 shows the chirp image interpolated using POCS. The POCS result is shown in (c). For comparison, the original chirp image is shown in (a) and the result using AH is shown in (b). POCS interpolates the region outside the diamond better than the AH algorithm. Aliasing is present only in the region near $(\omega_1, \omega_2) = (\pi, \pi)$. In the AH image, aliasing is present just outside the diamond region.

3.2.2 POCS - Red and Blue Images

POCS makes two observations about CFA interpolation to generate the red and blue images. First, measured values should be passed directly to the output. Second, if the red, green, and blue images are divided into subbands by a filterbank, the high-frequency subbands in

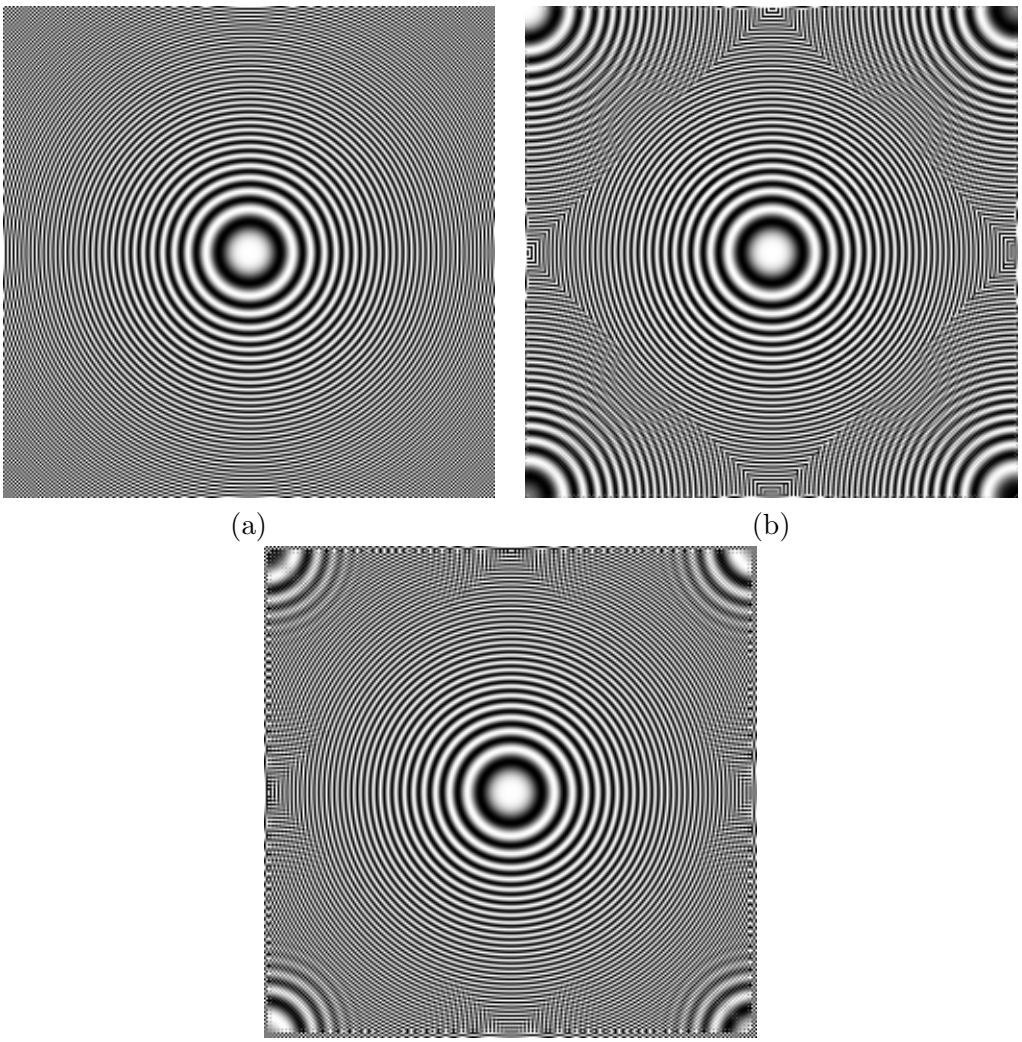


Figure 35: Chirp image results with green interpolation. (a) Original image. (b) AH. (c) POCS.

the three images should be similar. An iterative scheme is developed to enforce these two observations.

To initialize the system, bilinear interpolation is used to compute initial estimates for the red and blue images. These estimates, along with the interpolated green image, are divided into subbands using the same filterbank structure as the green system (see Figure 33). The low-frequency subband from the red image is combined with the three high-frequency subbands from the green image using the synthesis filterbank to generate the output red image. For the blue image, the output is generated by combining the blue low-frequency subband with the green high-frequency subbands. In this new image, red pixels

are replaced by the original red samples and blue pixels are replaced by the original blue samples. This image is again divided into subbands to perform another iteration. This iteration is repeated until the image converges, typically 3-5 times.

Without going into detail, it is seen that the green image is providing high-frequency information to the red and blue images. This improves the edge response in the output images. However, nothing in this algorithm explicitly works to cancel aliasing. Yet, as seen in Figure 36, this algorithm does a good job interpolating the chirp image with minimal aliasing. The only region where AH outperforms POCS is in the high-frequency regions along the horizontal and vertical axes. In the AH result, these areas have no aliasing coloring the image inside the diamond. However, the POCS image has a slight coloring inside the diamond region. In the regions outside the diamond, the AH result shows more aliasing than the POCS result.

3.3 Summary

This chapter focused on two common algorithms for CFA interpolation, AH and POCS. Both algorithms use edge-directed interpolation to interpolate the green image. In the case of POCS, we showed how this allows for nearly complete cancellation of aliasing in the green image. Also, both algorithms combine information from the red, green, and blue images. This is the main reason for the good results. By combining information, the algorithms are able to reduce aliasing and improve the overall system response.

These two algorithms are used as references in the remainder of this thesis. The next chapter develops two systems for interpolating the green image using ideas from AH and POCS.

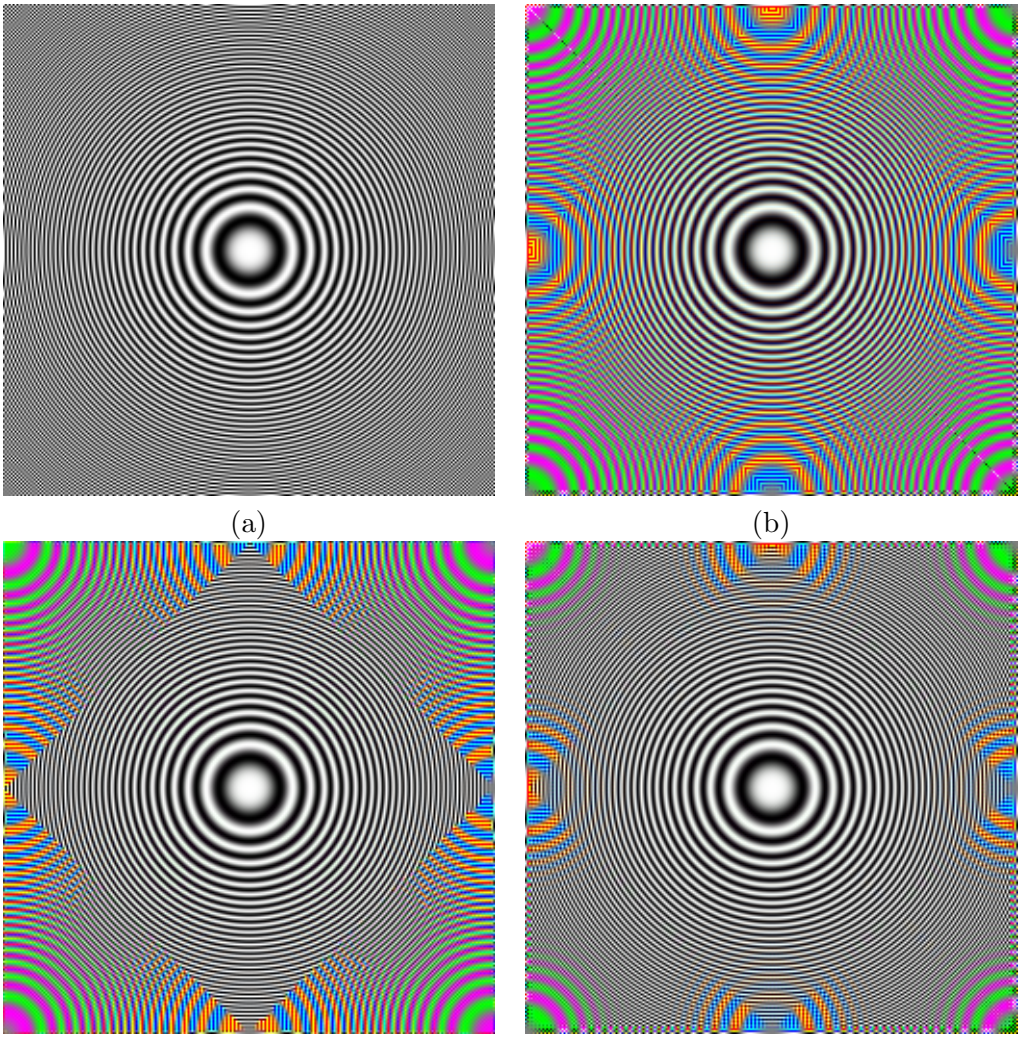


Figure 36: Color interpolation results for chirp image. (a) Original image. (b) Bilinear interpolation. (c) AH. (d) POCS.

CHAPTER IV

COLOR FILTER ARRAY INTERPOLATION - GREEN IMAGE

With the Bayer CFA, the green image is sampled on a quincunx sampling grid and the red and blue images are sampled on rectangular sampling grids. Therefore, the green image requires a different interpolation algorithm from the red and blue images. It is known from sampling theory that when an image is sampled with a quincunx sampling grid, an exact copy of the image can be reconstructed from its samples if the original image is bandlimited to a diamond-shaped region, as shown in Figure 37. To obtain this exact copy, an ideal

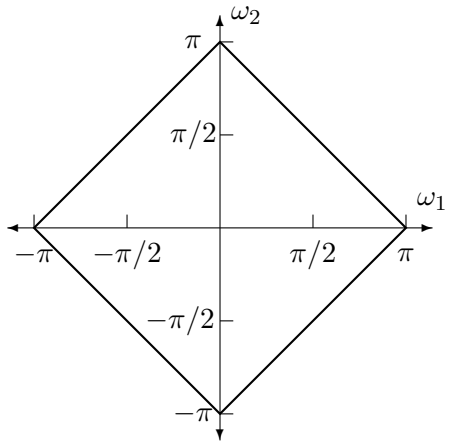


Figure 37: To avoid aliasing after sampling a signal with a quincunx sampling grid, a signal must be bandlimited to the diamond-shaped region.

diamond-shaped filter must be applied to the samples, but ideal interpolation filters are impractical because of their infinite length. Therefore, we design an approximate diamond-shaped filter using standard filter design techniques. For example, bilinear interpolation

uses a low-order approximation,

$$H_g^{bilin} = \begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 1 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}. \tag{64}$$

The 2-D frequency response of this filter is shown in Figure 38. From the slice along the horizontal frequency axis, we can see that the filter does not have a flat passband response and the transition band is very wide, allowing significant energy in the stopband to pass to the output. As we have seen, when this filter is applied to an example image, numerous artifacts result.

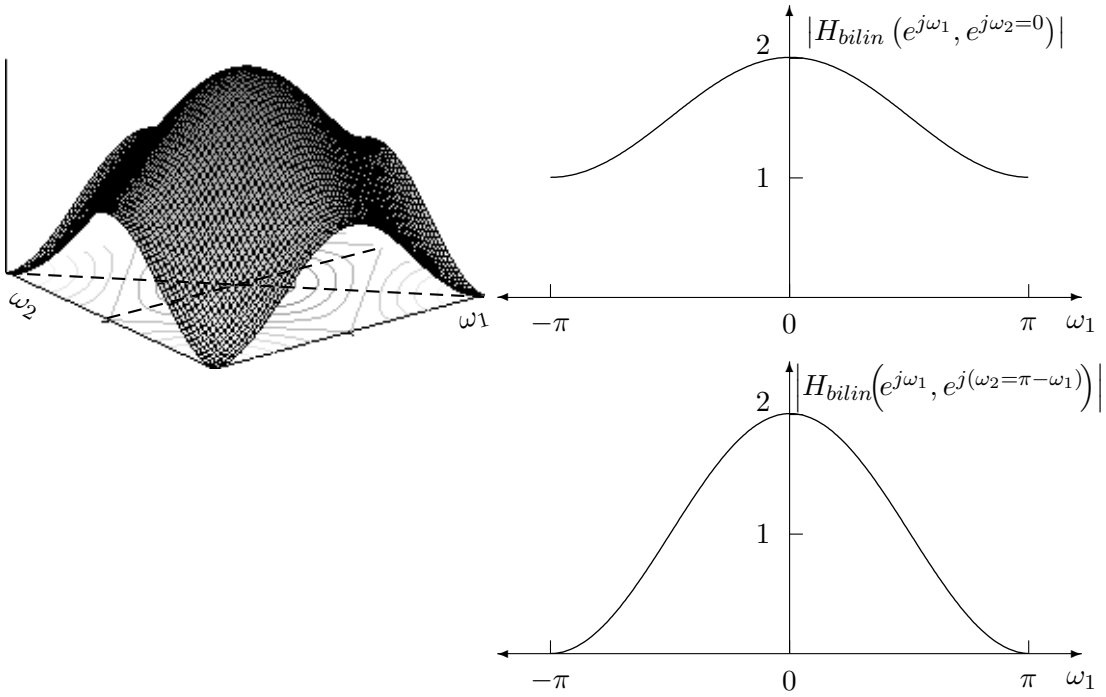


Figure 38: Bilinear interpolation uses a low-order approximation of a diamond-shaped filter on the green image.

A better approximation of the ideal interpolation filter can be designed by increasing the filter order. Shown in Figure 39 is the filter response of a 23x23 diamond-shaped filter. Figure 40 shows the result image when the 23x23 filter is applied to the green samples. Although the filter frequency response is much closer to the ideal, the result images show similar artifacts as the bilinear interpolation filter. The railing at the top of the lighthouse

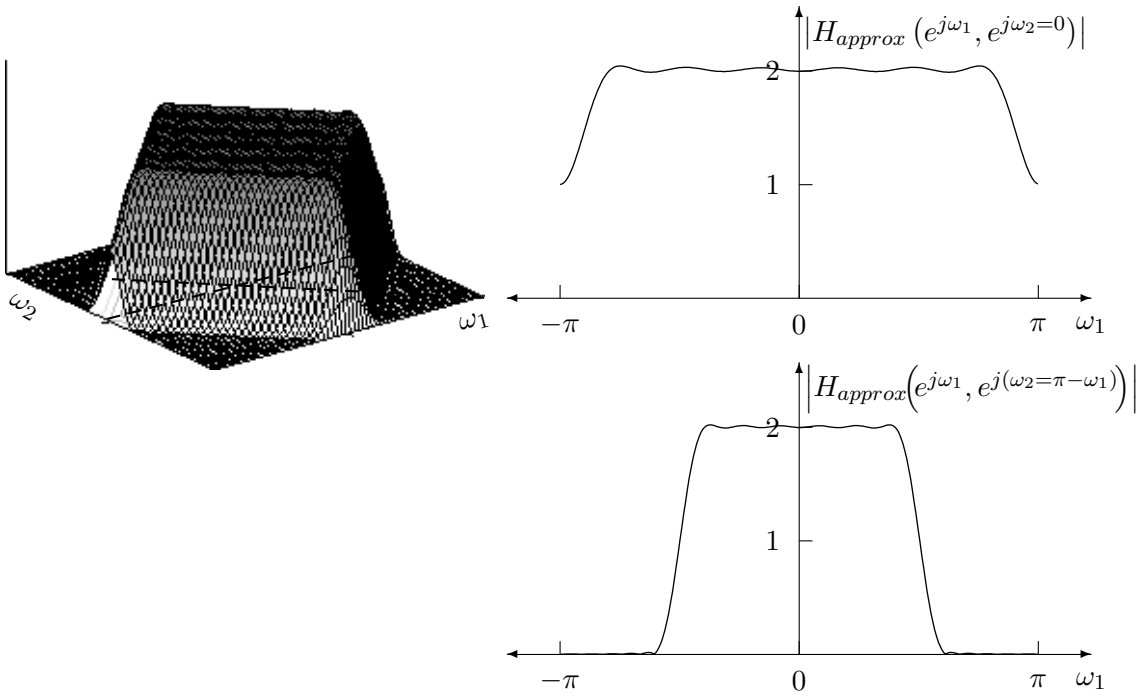


Figure 39: Frequency response of a 23x23 diamond-shaped interpolation filter.

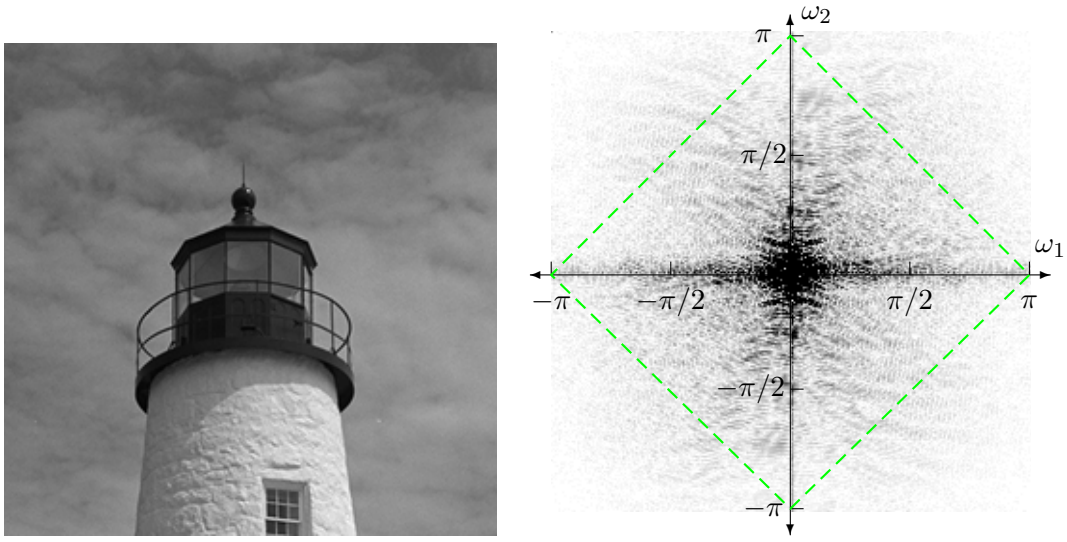
shows aliasing artifacts. There is also evidence of aliasing along the fence and on the siding of the building. In Figure 41, the spectrum is shown for the region around the rails of the lighthouse. Portions of the spectrum extend outside of the diamond-shaped Nyquist region, showing that aliasing will result in the interpolated image.

When aliasing is present, using a single interpolation filter on the green image is insufficient to produce a good image. Therefore, this chapter introduces two new methods to interpolate the green image. Both use the correlation among the red, green, and blue images. Like the Adams and Hamilton (AH) algorithm, a correction signal from the red and blue images is added to the interpolated green image to reduce aliasing problems.

4.1 Aliasing Correction Filter

Edge-directed interpolation algorithms require a decision to be made for each green pixel that is estimated. The algorithm decides whether a horizontal or vertical interpolator is used for each pixel. This class of program does not run efficiently on a Digital Signal Processing (DSP) microprocessor. Because the algorithm may change the interpolation filter from a

Figure 40: A 23x23 interpolation filter is used to interpolate the green image of the *Lighthouse* example. Aliasing can be seen along the rails of the lighthouse, along the fence, and on the siding of the building.



(a) (b)

Figure 41: Spectrum of rails region of *Lighthouse* image. (a) Rails region of image. (b) Spectrum of rails region.

horizontal filter to a vertical filter at each pixel, the processor pipeline is never filled and the program runs with a greater amount of overhead. For this reason, a system that can achieve the same results as AH without performing the directional decision is desired.

In Chapter 3, we showed that the AH algorithm used information from the red, green, and blue images to interpolate the green image. The correlation between the images allows for an improved interpolated green image. In that chapter, it was shown that the green image provides the low-frequency information to the output image. The interpolation filter had a non-flat passband and allowed energy from the stopband into the output image. Because the input to this interpolation filter is a sampled image on the 1/2-rate grid, a shifted copy of the original spectrum lies in the stopband region of the interpolation filter. Thus, these stopband imperfections in the interpolation filter lead to aliasing artifacts in the output image. Information from the red and blue images was used to reduce these aliasing artifacts.

This section provides a general system to achieve the same results as AH. This new system eliminates the edge-directed interpolation, making it more efficient to implement on DSP processors. In Figure 22, the block diagram for the AH algorithm is presented. This

block diagram assumes that the interpolation direction has been decided. Therefore, the block diagram is a one-dimensional system. In this simplified system, a lowpass filter was applied to the green image and a correction filter was applied to the red or blue image. This can be expanded to a two-dimensional system that does not use edge-directed interpolation. In this system, a lowpass interpolation filter is designed for the green image. Given the interpolation filter coefficients, the coefficients for a correction filter are optimized to correct the passband roll-off and the aliasing allowed by an imperfect stopband.

Figure 42 introduces the block diagram for the green interpolation system. Recall the sampling equations for the red, green, and blue images.

$$G_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}G\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{2}G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) \quad (65)$$

$$R_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{4}R\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{4}R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + \frac{1}{4}R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) \quad (66)$$

$$B_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{4}B\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{4}B\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - \frac{1}{4}B\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}B\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) \quad (67)$$

Substituting these equations into the block diagram leads to an equation for the estimated green image,

$$\begin{aligned}\hat{G}\left(e^{j\omega_1}, e^{j\omega_2}\right) = {} & \frac{1}{2}\left[G\left(e^{j\omega_1}, e^{j\omega_2}\right) + G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right]H_1\left(e^{j\omega_1}, e^{j\omega_2}\right) \\ & + \frac{1}{4}\left[R\left(e^{j\omega_1}, e^{j\omega_2}\right) - R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right]H_2\left(e^{j\omega_1}, e^{j\omega_2}\right) \\ & + \frac{1}{4}\left[B\left(e^{j\omega_1}, e^{j\omega_2}\right) + B\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - B\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - B\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right]H_2\left(e^{j\omega_1}, e^{j\omega_2}\right). \quad (68)\end{aligned}$$

Applying the perfect correlation assumption, some terms can be cancelled. Assuming that the red and blue images have the same frequency content in the passband of H_2, the $\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)$ and $\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$ terms in the red and blue signals can be removed because these terms have opposite signs.

$$\hat{G}\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}\left[G\left(e^{j\omega_1}, e^{j\omega_2}\right) + G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right]H_1\left(e^{j\omega_1}, e^{j\omega_2}\right)$$

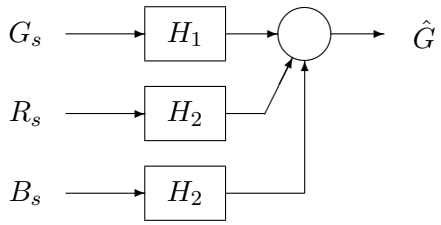


Figure 42: Block diagram for the aliasing cancellation green interpolation system.

$$+\frac{1}{4}\Big[R\left(e^{j\omega_1},e^{j\omega_2}\right)+B\left(e^{j\omega_1},e^{j\omega_2}\right)- R\Big(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\Big)-B\Big(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\Big)\Big]H_2\Big(e^{j\omega_1},e^{j\omega_2}\Big). \quad (69)$$

This system has a similar structure as the AH system in (34). The correction terms from the red and blue images add to the green image for the $\left(e^{j\omega_1},e^{j\omega_2}\right)$ term, but subtract from the green image for the $\left(e^{j(\omega_1-\pi)},e^{j(\omega_2-\pi)}\right)$ term. This allows for the alias cancellation along with correction for the interpolation roll-off.

4.1.1 Filter Design

Any standard filter design technique can be used to design the diamond-shaped interpolation filter, H_1. An example filter is designed by recognizing that a diamond-shaped filter is a rectangular filter rotated by 45°. A rectangular filter can be designed with a 1-D prototype lowpass filter. The rectangular filter is obtained by treating the prototype filter as a row filter and convolving it with the same filter as a column filter. The kernel for the rectangular filter is rotated to form the diamond-shaped filter. The design procedure is illustrated in Figure 43 for the 3x3 case. The same procedure is used to design larger filters for the interpolation system.

Given the diamond-shaped interpolation filter, the correction filter, H_2, is designed using an optimization procedure. This optimization has two goals:

1. Force the sum of $H_1\left(e^{j\omega_1},e^{j\omega_2}\right)$ and $H_2\left(e^{j\omega_1},e^{j\omega_2}\right)$ to have a flat passband.

2. Force the difference of $H_1\left(e^{j\omega_1},e^{j\omega_2}\right)$ and $H_2\left(e^{j\omega_1},e^{j\omega_2}\right)$ to be as close to zero as possible.

1. Generate a rectangular lowpass filter by convolving a 1-D prototype lowpass filter with its transpose.

$$\begin{bmatrix} h_1 & h_0 & h_1 \end{bmatrix} ** \begin{bmatrix} h_1 \\ h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} h_1^2 & h_0 h_1 & h_1^2 \\ h_0 h_1 & h_0^2 & h_0 h_1 \\ h_1^2 & h_0 h_1 & h_1^2 \end{bmatrix}$$

2. Rotate the rectangular filter in the frequency domain using the rotation matrix, $\mathbf{R}$, to form a diamond-shaped filter.

$$\begin{aligned} H(\omega) &= \sum_{\mathbf{n}} h[\mathbf{n}] e^{-j\omega^T \mathbf{n}} \\ H(\mathbf{R}\omega) &= \sum_{\mathbf{n}} h[\mathbf{n}] e^{-j\omega^T \mathbf{R}^T \mathbf{n}} \\ H(\mathbf{R}\omega) &= \sum_{\mathbf{m}} h[\mathbf{R}^{-T}\mathbf{m}] e^{-j\omega^T \mathbf{m}} \end{aligned}$$

$$\begin{aligned} \omega^T &= \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \\ \mathbf{n}^T &= \begin{bmatrix} n_1 & n_2 \end{bmatrix} \\ \mathbf{R} &= \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \\ \mathbf{R}^{-T} &= \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \end{aligned}$$

3. This last equation shows that a rotation in the frequency domain is equivalent to a rotation in the spatial domain. Points mapped to non-integer coordinates by the coordinate transformation are discarded.

$$\begin{bmatrix} 0 & h_1^2 & 0 \\ h_1^2 & h_0^2 & h_1^2 \\ 0 & h_1^2 & 0 \end{bmatrix}$$

EXAMPLE: Prototype filter

$$\begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix}$$

Rectangular lowpass filter

$$\begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix} ** \begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

Rotate to form diamond-shaped filter, discarding points mapped to non-integer locations

$$\begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

Figure 43: The example interpolation filter is designed by applying a 1-D prototype filter separably. The resulting rectangular filter is rotated to form a diamond-shaped filter.

The first goal tries to eliminate the effects of the roll-off of the green interpolation filter. The second goal attempts to reduce aliasing resulting from the $\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$ terms.

A least squares optimization was used to compute the filter coefficients. According to the first goal, the sum of H_1 and H_2 should be flat inside the diamond-shaped region and the magnitude of the combination filter should be 2. Therefore, the first error measure is

$$\xi_1 = \int\int_{\mathbf{R}_1} \left[2 - H_1\left(e^{j\omega_1}, e^{j\omega_2}\right) - H_2\left(e^{j\omega_1}, e^{j\omega_2}\right)\right]^2 d\omega_1 d\omega_2 \tag{70}$$

where $\mathbf{R_1}$ defines the diamond-shaped region in Figure 37. The second error measure is

$$\xi_2 = \int\int_{\mathbf{R}_2} \left[H_1\left(e^{j\omega_1}, e^{j\omega_2}\right) - H_2\left(e^{j\omega_1}, e^{j\omega_2}\right)\right]^2 d\omega_1 d\omega_2. \tag{71}$$

where $\mathbf{R_2}$ defines the region outside the diamond. The total error measure for the optimization procedure is given by

$$\xi = \alpha_1\xi_1 + \alpha_2\xi_2 \tag{72}$$

with

$$\alpha_1 + \alpha_2 = 1.$$

In practice, $\alpha_1 = \alpha_2 = 1/2$ to equally weight the passband error and stopband error and the integrals are computed by summation over a sufficiently dense set of points inside the region of integration.

Figure 44 shows the frequency responses of the interpolation filter and the correction filter for filter sizes of 5, 9, 13, and 17. In all four cases, the correction filter is a 2-D bandpass filter with passband around the diamond-shaped interpolation filter. This is similar to the 1-D bandpass correction filter used in the edge-directed algorithm by Adams and Hamilton.

4.1.2 Results

In Figure 45, these filters were applied to the *Lighthouse* example image. The original green image is compared to the interpolated images and several regions of the image are shown. Aliasing can be seen in the fence region of the image in Figure 45. The 5x5 filter set shows more aliasing than the larger extent filters. On the siding of the building, aliasing can be seen in the 5x5 result, but not in the results from the other filter sets. The longer filters

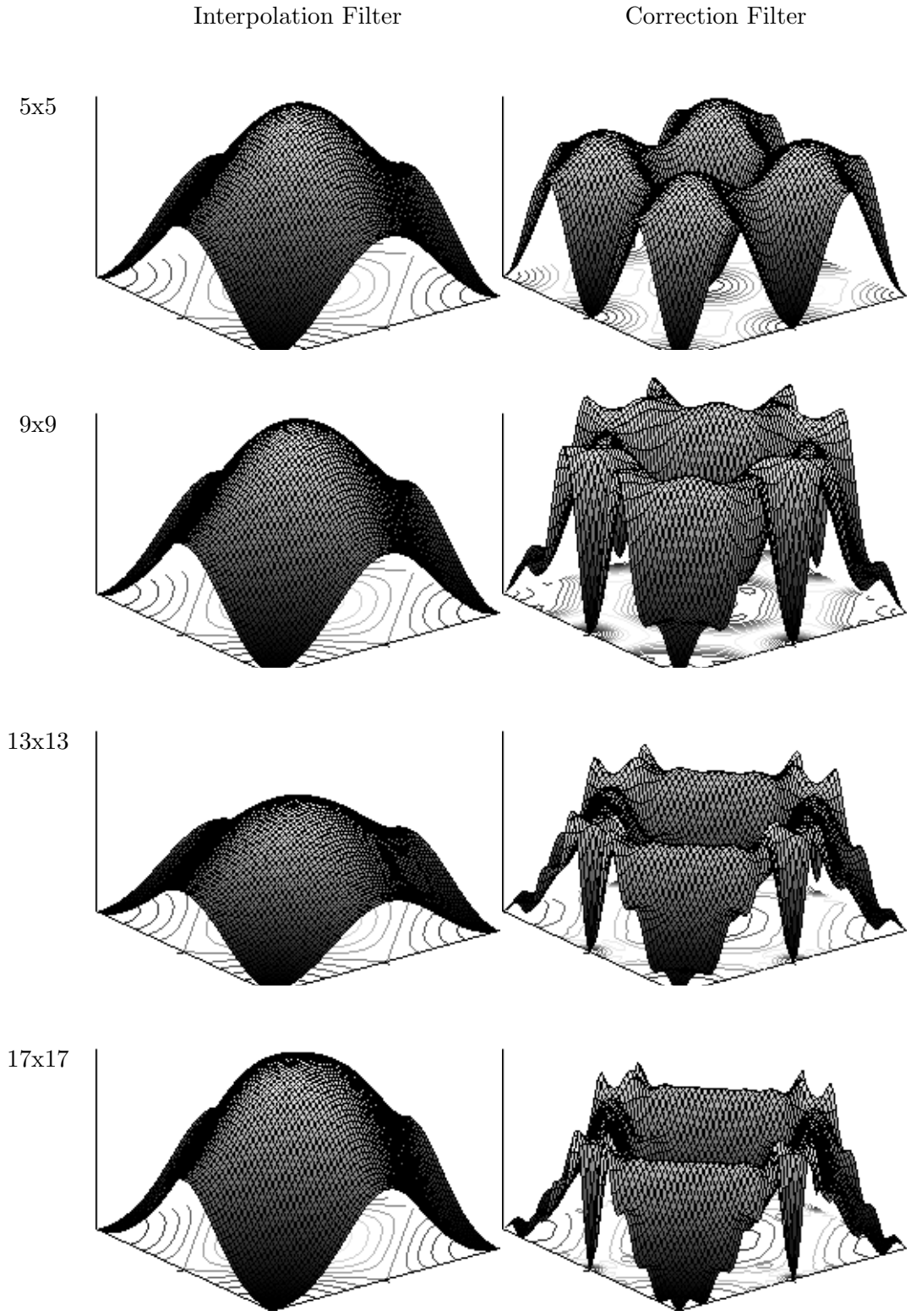


Figure 44: Filter responses for the 5x5, 9x9, 13x13, and 17x17 filters. The surface plots on the left show the interpolation filter and the plots on the right show the correction filters.

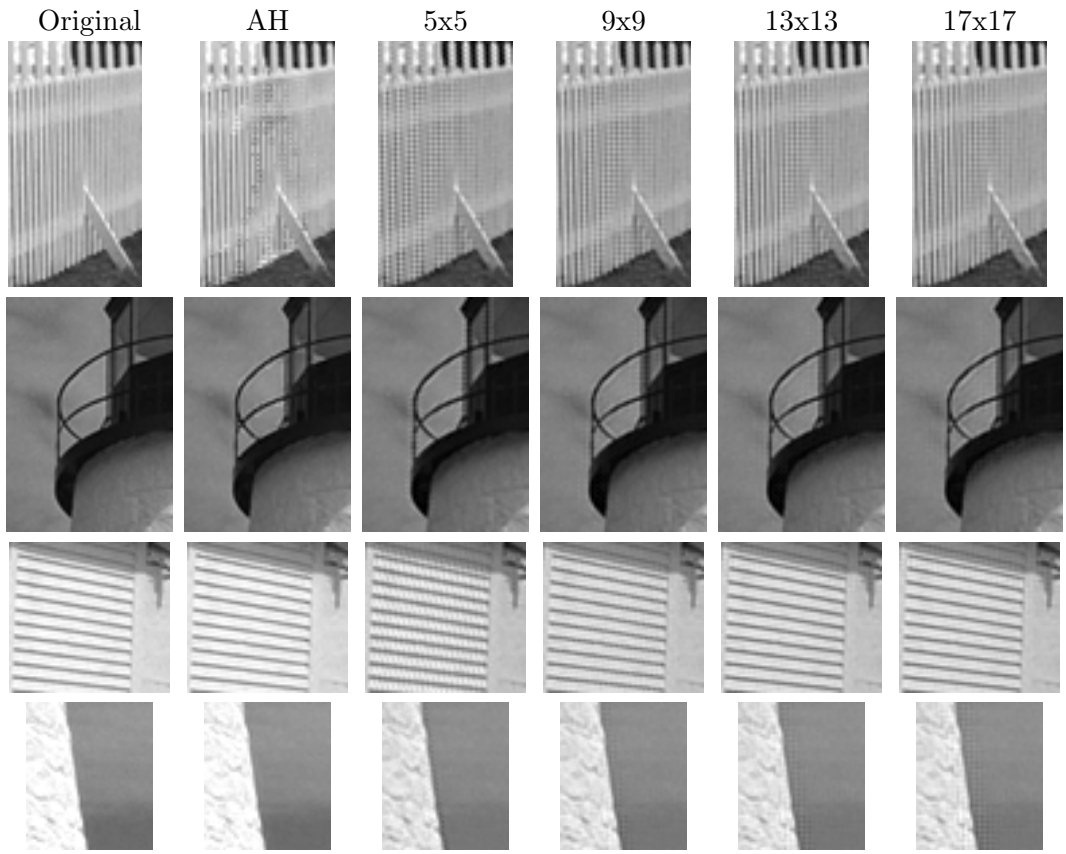


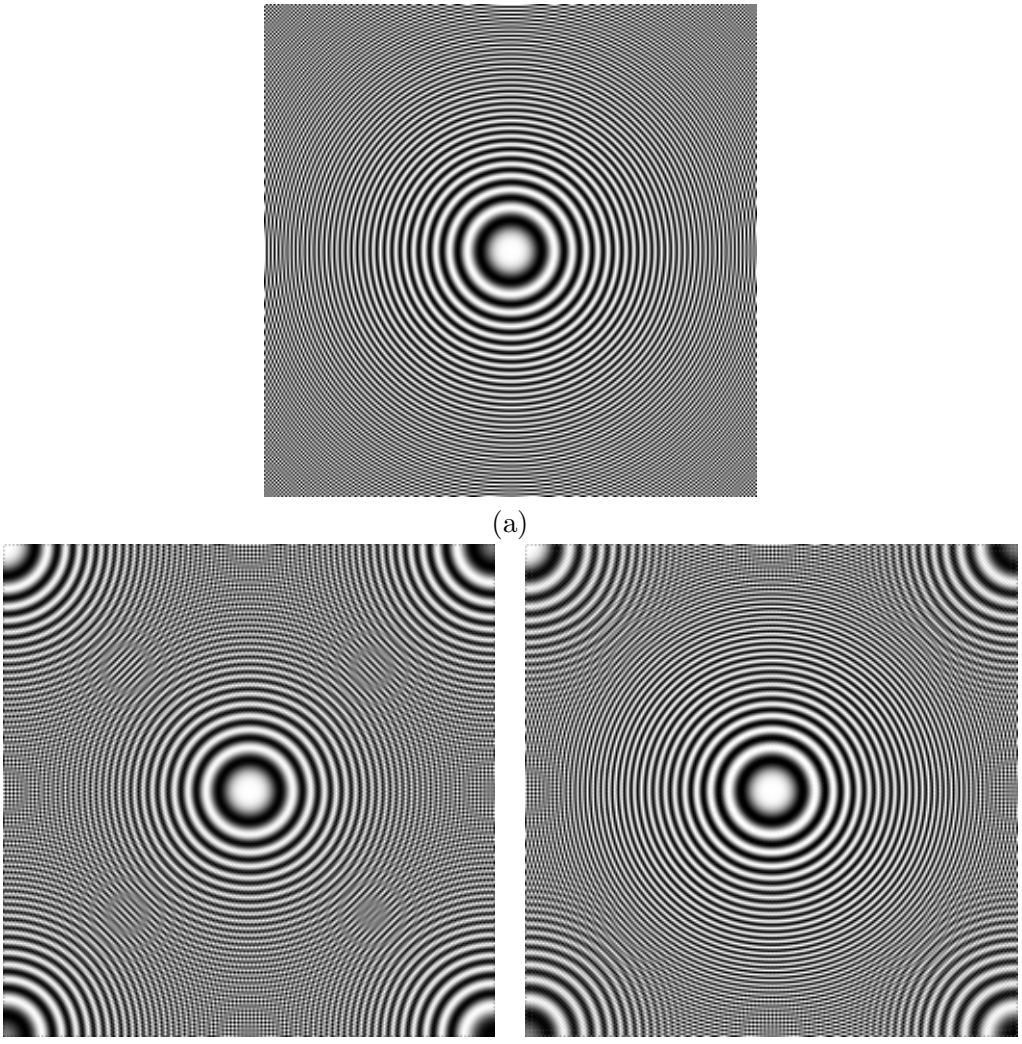
Figure 45: The green image of *Lighthouse* was interpolated with 5x5, 9x9, 13x13, and 17x17 filters. The original image for each case is shown in the leftmost image.

provide a better system response. However, the edge response of the longer filters sometimes leads to visual artifacts. This is seen along the border of the lighthouse. The result images of the 13x13 and 17x17 filter sets show apparent edge artifacts due to the edge response of the system. Figure 46 uses the green image of *Lighthouse* to show the progression of using the interpolation filter, then adding the signal from the correction filter. This shows how the correction filter reduces aliasing, for example around the railing on the lighthouse. This correction is illustrated better with the chirp image in Figure 47. The image in Figure 47(b) was interpolated using only H_1 on the green samples. The mid-frequencies show a large amount of aliasing. The corrected image in (c) shows how this aliasing is reduced inside the diamond-shaped Nyquist region.

Table 1 shows the MSE results from applying the four filter sets to the entire *Kodak* color image database. In most cases, the results of the 9x9 filters give the lowest mean

(a) (b) (c)

Figure 46: Using the 9x9 interpolation and correction filters, the green image of *Lighthouse* was interpolated with just the interpolation filter in (b) and the output of the correction filter was added in (c). (a) shows the original image.



(a)

(b) (c)

Figure 47: The chirp image shows the aliasing reduction using the correction filter. (a) Original image. (b) Interpolated green image without correction. (c) Correction terms added from the red and blue images.

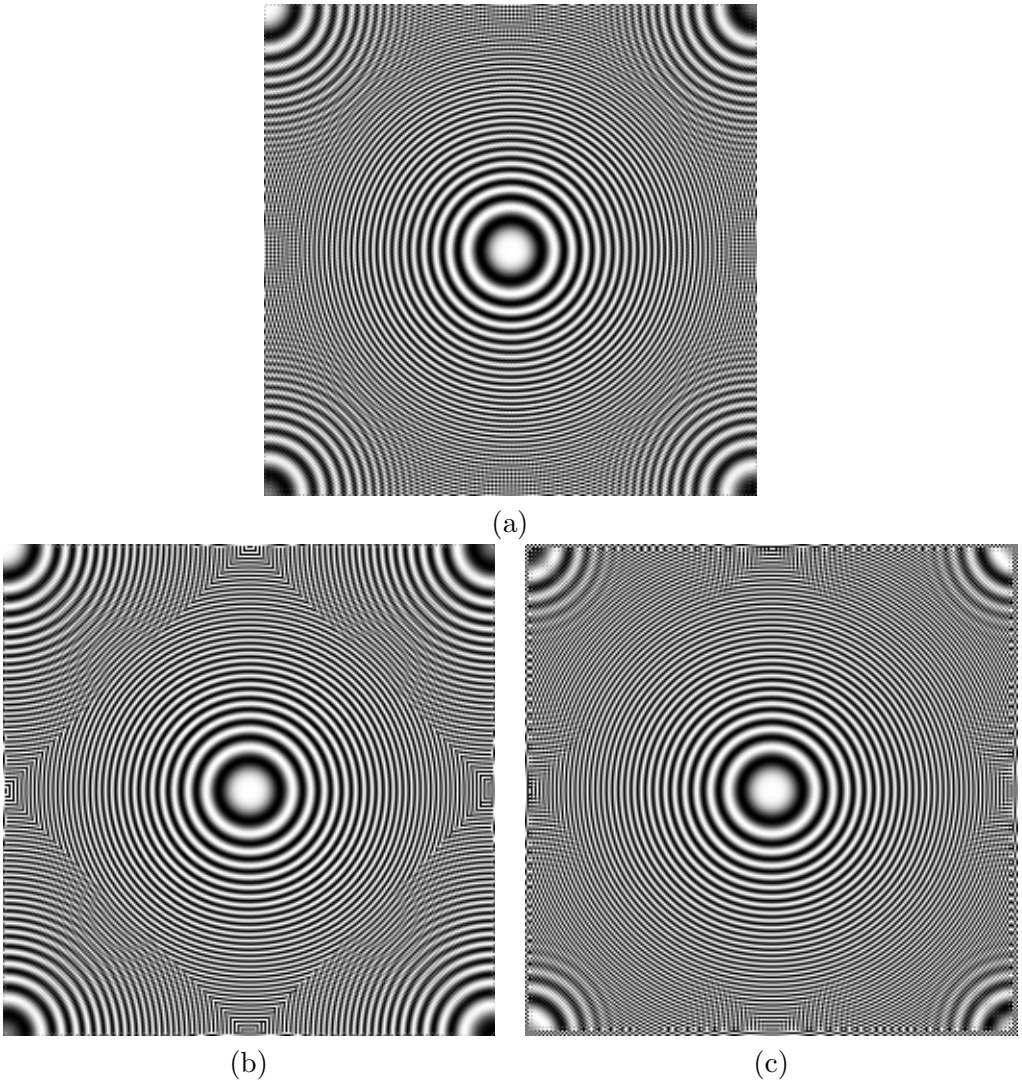
Table 1: MSE results from *Kodak* color image database. Filter sets of 5x5, 9x9, 13x13, and 17x17 extent were used. The results from AH are also included for comparison.

Image	*AH*	*5x5*	*9x9*	*13x13*	*17x17*
1	22.6694	19.8255	13.817	13.1969	16.4155
2	5.2988	5.56897	6.66386	9.22549	11.7247
3	3.98055	3.9161	5.14359	7.64189	10.6291
4	5.99072	5.16002	5.94531	8.28358	10.5814
5	17.3563	18.7842	17.4482	21.924	29.0953
6	16.9749	14.804	9.65742	9.5159	11.5403
7	3.99161	5.56352	5.80836	7.69694	10.0867
8	30.2512	39.938	24.5388	22.7665	26.4701
9	4.81386	6.04376	5.57504	7.17038	9.38412
10	4.87273	4.9668	4.41072	5.41927	7.03634
11	13.2106	11.8284	10.1166	11.633	14.9559
12	3.95468	4.48275	3.62523	4.24911	5.36666
13	57.5927	38.9867	33.2138	32.3229	40.7614
14	12.786	13.5679	21.9201	33.8147	43.5295
15	7.28241	5.66201	6.35189	8.10379	10.3395
16	7.39154	6.9093	4.38703	4.40392	5.52963
17	7.97153	6.80963	6.41104	6.83569	8.47074
18	18.6834	15.2005	15.4942	17.4758	22.4016
19 (*Lighthouse*)	9.62141	14.8339	8.7439	8.46166	9.95218
20	6.80694	7.22263	6.78142	8.65986	11.4728
21	15.5258	13.3829	10.7067	11.5138	14.6306
22	11.2832	10.5423	10.0406	11.7152	14.4109
23	3.14131	3.87705	5.26322	8.96087	13.4056
24	28.6043	21.554	19.1238	20.2736	24.542

square error. This matches the observation that the 13x13 and 17x17 filters lead to visible edge artifacts.

4.2 Filterbank Method of Green Interpolation

The chirp image shows how each part of the spectrum is handled by each of the algorithms. Figure 48 shows the chirp result for (a) the aliasing correction filter, (b) AH, and (c) POCS. The results for AH and the aliasing correction system are similar. The AH algorithm reconstructs the high horizontal and high vertical frequencies better than the aliasing correction system. Both have a boundary around the diamond-shaped Nyquist region where inside the boundary, aliasing is removed, but immediately outside the boundary, aliasing is present. The POCS image has no such boundary. It reconstructs frequencies far outside the diamond



(a)

(b) (c)

Figure 48: Comparison of the correction filter to the reference algorithms, AH and POCS. (a) 9x9 Correction filter. (b) AH. (c) POCS.

without aliasing. This section presents a second method of interpolating the green image that attempts to achieve this same result.

4.2.1 Background

Suppose two perfectly-correlated one-dimensional signals, $G\left(e^{j\omega}\right)$ and $R\left(e^{j\omega}\right)$, are sampled on interleaved one-dimensional sampling grids. The sampled signals have the following Fourier transforms,

$$G_s\left(e^{j\omega}\right) = \frac{1}{2}G\left(e^{j\omega}\right) + \frac{1}{2}G\left(e^{j(\omega-\pi)}\right) \tag{73}$$

$$R_s\left(e^{j\omega}\right) = \frac{1}{2}R\left(e^{j\omega}\right) - \frac{1}{2}R\left(e^{j(\omega-\pi)}\right). \quad (74)$$

In this example, the original signal, $G\left(e^{j\omega}\right)$, is reconstructed by applying the filter $H_1\left(e^{j\omega}\right)$ to $G_s\left(e^{j\omega}\right)$ and the filter $H_2\left(e^{j\omega}\right)$ to $R_s\left(e^{j\omega}\right)$ to give

$$\begin{aligned}\hat{G}\left(e^{j\omega}\right) = & \frac{1}{2}H_1\left(e^{j\omega}\right)G\left(e^{j\omega}\right) + \frac{1}{2}H_1\left(e^{j\omega}\right)G\left(e^{j(\omega-\pi)}\right) + \\ & \frac{1}{2}H_2\left(e^{j\omega}\right)R\left(e^{j\omega}\right) - \frac{1}{2}H_2\left(e^{j\omega}\right)R\left(e^{j(\omega-\pi)}\right). \quad (75)\end{aligned}$$

The idea is to use the correlation between the original two signals to rebuild the original signal, $G\left(e^{j\omega}\right)$. Substitute $R(e^{j\omega}) = G(e^{j\omega}) + k_r\delta(\omega)$ where k_r is constant into (75) to get

$$\begin{aligned}\hat{G}\left(e^{j\omega}\right) = & \frac{1}{2}H_1\left(e^{j\omega}\right)G\left(e^{j\omega}\right) + \frac{1}{2}H_1\left(e^{j\omega}\right)G\left(e^{j(\omega-\pi)}\right) + \\ & \frac{1}{2}H_2\left(e^{j\omega}\right)G\left(e^{j\omega}\right) + \frac{1}{2}H_2\left(e^{j\omega}\right)k_r\delta\left(\omega\right) - \\ & \frac{1}{2}H_2\left(e^{j\omega}\right)G\left(e^{j(\omega-\pi)}\right) - \frac{1}{2}H_2\left(e^{j\omega}\right)k_r\delta\left(\omega-\pi\right). \quad (76)\end{aligned}$$

It is safe to assume that $H_2\left(e^{j\omega}\right) = 0$ near $\omega = 0$ because $G\left(e^{j\omega}\right)$ is different from $R\left(e^{j\omega}\right)$ near $\omega = 0$. Since $H_2\left(e^{j\omega}\right)$ is applied to $R\left(e^{j\omega}\right)$ and added to the green result, $H_2\left(e^{j\omega}\right)$ should be zero near $\omega = 0$ to avoid changing the DC value of the output green signal. Also, because $H_2\left(e^{j\omega}\right)$ is applied to $R\left(e^{j(\omega-\pi)}\right)$ and added to the green result, $H_2\left(e^{j\omega}\right)$ should be zero near $\omega = \pi$. These assumptions say that $H_2\left(e^{j\omega}\right)k_r\delta\left(\omega\right) = 0$ and $H_2\left(e^{j\omega}\right)k_r\delta\left(\omega-\pi\right) = 0$. Thus, (76) simplifies to

$$\begin{aligned}\hat{G}\left(e^{j\omega}\right) = & \frac{1}{2}H_1\left(e^{j\omega}\right)G\left(e^{j\omega}\right) + \frac{1}{2}H_1\left(e^{j\omega}\right)G\left(e^{j(\omega-\pi)}\right) + \\ & \frac{1}{2}H_2\left(e^{j\omega}\right)G\left(e^{j\omega}\right) - \frac{1}{2}H_2\left(e^{j\omega}\right)G\left(e^{j(\omega-\pi)}\right). \quad (77)\end{aligned}$$

The system of equations that needs to be solved is:

$$\begin{aligned}H_1\left(e^{j\omega}\right) + H_2\left(e^{j\omega}\right) &= 2 \\ H_1\left(e^{j\omega}\right) - H_2\left(e^{j\omega}\right) &= 0. \quad (78)\end{aligned}$$

This system has a unique solution that both filters should be identity filters, $H_1\left(e^{j\omega}\right) = H_2\left(e^{j\omega}\right) = 1$. This solution contradicts the two assumptions made in the simplification that $H_2\left(e^{j(\omega=0)}\right) = 0$ and $H_2\left(e^{j(\omega=\pi)}\right) = 0$. This solution does give insight to an approximate solution appropriate for the problem to be solved.

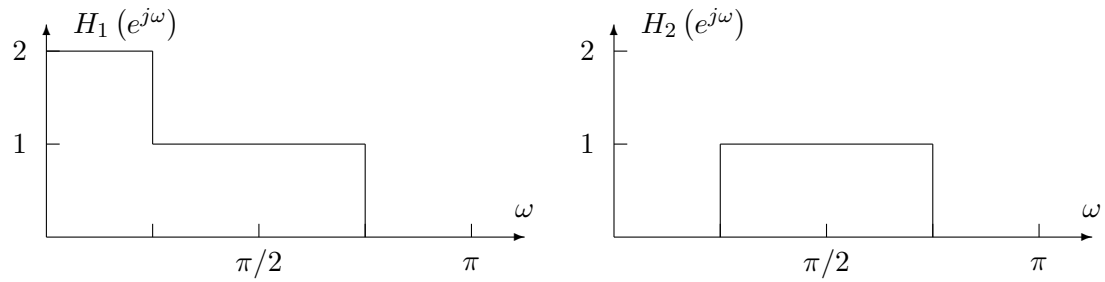


Figure 49: An approximate solution to the one-dimensional problem. Note that the filter responses are symmetric and only the positive frequencies are shown.

The problem areas in the spectrum for the exact solution are those areas where the original signals differ. In particular, this would be near the DC region since only in a non-colored region would the two signals have the same average value. In the one-dimensional case, after sampling, this problem area would be replicated at $\omega = \pi$. Thus, in these areas, the filters should deviate from the exact solution. Figure 49 shows an approximate solution appropriate for the one-dimensional problem. The main limitation of this system is the resulting artifacts when the original signal contains energy in the range $3\pi/4 \leq \omega < \pi$. The aliased copy of this energy is passed by $H_1\left(e^{j\omega}\right)$ in the $G\left(e^{j(\omega-\pi)}\right)$ signal, but cannot be cancelled by the $R\left(e^{j(\omega-\pi)}\right)$ term since $H_2\left(e^{j\omega}\right)$ is 0 in this region.

This solution is easily extended to two dimensions and to the problem of CFA interpolation. The solution is shown in Figure 50. In CFA interpolation, two signals, the red and blue samples, can be used as correction signals. Both signals are combined evenly to form the correction signal. The extension to two dimensions requires recognizing where all of the spectral copies are located in the spectrum. The filter used on the green image should have a magnitude of 2 in the region around $(\omega_1, \omega_2) = (0, 0)$ and a magnitude of 0 in the region near (π, π). These are the locations of the copies of the original spectrum. At other points in the spectrum, the green filter has a magnitude of 1. The correction filter used on the red and blue images needs to have a magnitude of 0 in the regions near $(0, 0)$, $(\pi, 0)$, $(0, \pi)$, and (π, π) because the rectangular sampling grid used on the red and blue images creates spectral copies at these locations. Since the original signals have different average values, these regions would be expected to have little correlation between the signals. The

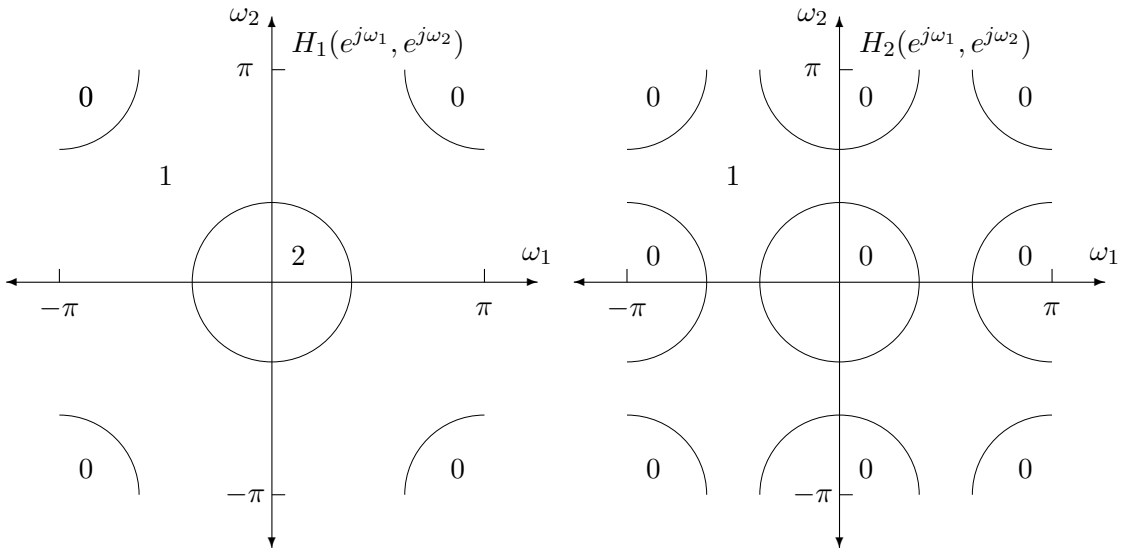


Figure 50: An approximate solution for interpolating the green image.

correction filter has a magnitude of 1 at other points in the spectrum.

Both the sum and the difference of these two filters are important characteristics of the system. The sum of the two filters describes the overall system response. A system that cannot reproduce high frequency components gives a soft, or blurred, image. The difference between the two filters shows the system's response to the aliasing terms. Figure 51 reveals the tradeoff between system response and aliasing rejection. To perfectly reconstruct the original signal, the sum of the responses should be a constant value of 2 and the difference of the responses should be 0. This is described by the system of equations in (78). The ideal solution perfectly passes the original signal while completely rejecting the aliasing terms. In the approximate solution, $H_1 + H_2$ matches the ideal solution except in the high-frequency regions. Along the axes, where horizontal and vertical components exist, the system attenuates the energy. This causes some blurring of details of sharp horizontal and vertical edges. The system completely rejects energy near the corners of the spectrum. This is the region where all three signals contain aliasing components from the strong DC component in the original image. Since the DC components in the three signals are uncorrelated, no helpful information can be drawn to restore information in this part of the spectrum.

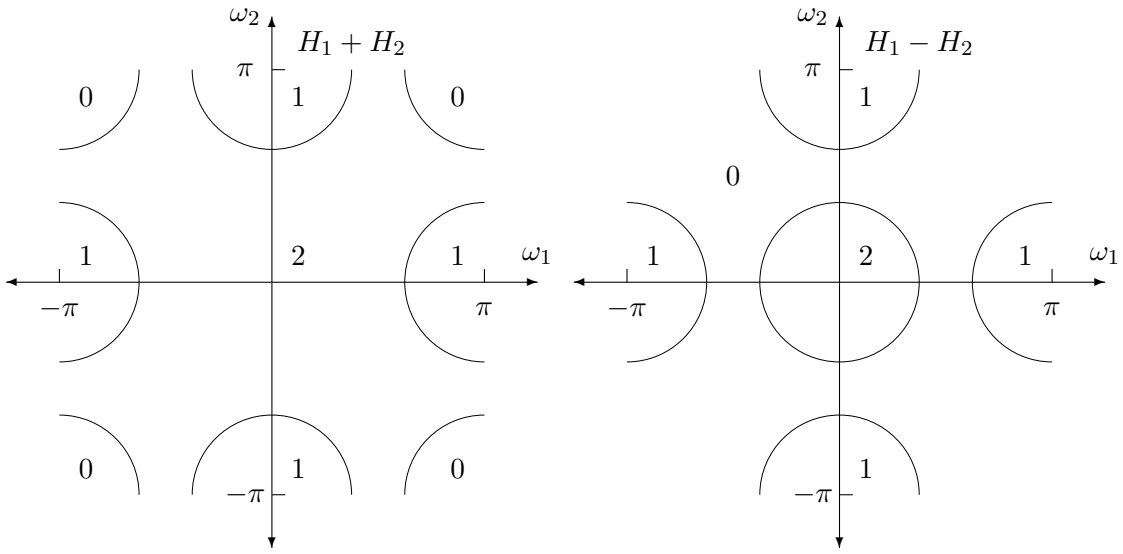


Figure 51: The sum and difference of the interpolation filters show the overall system response and the system's response to aliasing.

The response of $H_1 - H_2$ shows where the system is likely to suffer from aliasing artifacts. In the 1-D system description, (75), this difference filter is applied to the shifted terms, $G\left(e^{j(\omega-\pi)}\right)$ and $R\left(e^{j(\omega-\pi)}\right)$. This is the same in the 2-D system where the difference filter is applied to the $\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$ terms. For the approximate solution, energy is passed at the high-frequency portions of the horizontal and vertical axes. Thus, aliasing is present when the original image contains horizontally and vertically aligned features with sharp edges or repeated patterns with dense spacing. Because this is a quincunx grid, the horizontal features lead to vertical artifacts and vertical features lead to horizontal artifacts. The aliasing region near the origin in the spectrum generates artifacts when the original signal contains features at a 45° angle. Energy from these features are aliased near the DC-point of the spectrum, where the approximate solution does not reject aliasing. These limitations of the system suggest a tradeoff between reducing the spectral area of possible aliasing and introducing new artifacts by using portions of the spectrum where the original signals are different.

4.2.2 Implementation

To implement a system that achieves the approximate solution, a set of filters is needed. The filter used on the green image needs to have a magnitude response that matches the response shown for H_1 in Figure 50. It has a magnitude of 2 near the origin to cancel the 1/2 factor from the sampling process. It also completely blocks frequencies near (π, π) in the spectrum since the sampling process replicates the original spectrum at this location. The strongest components in the original image lie near the origin, so the replicated spectrum will have strong components near (π, π) that need to be blocked. In the remainder of the spectrum, the filter should have a magnitude of 1. The filter for the red and blue images corrects the green filter and should match the response shown for H_2 in Figure 50. These images are sampled on a rectangular grid, so the spectrum is replicated at $(\pi, 0)$, $(0, \pi)$, and (π, π). Thus, the correction filter should block frequencies in these regions. Also, the average values of the red, green, and blue images are only the same in gray portions of the image. So that the correction filter does not change the average value of the green image, it needs to block the region near the origin in the spectrum. In the rest of the spectrum, the correction filter should have a magnitude of 1. An analysis filterbank will be used to build this filter.

Filterbanks are a useful tool in signal processing to divide a signal into frequency-selective bands. Two-dimensional filterbanks can be used for image processing to achieve the same result. Since the system is described with respect to various regions of the spectrum, a filterbank system is a useful tool to implement this system. A two-level tree-structured analysis filterbank is shown in Figure 52. This structure divides the sampled images into sixteen subimages as shown in Figure 53.

Each of the subimages represents a different part of the spectrum. To match the approximate solution in Figure 50, each of the subimages is scaled with a factor of 0, 1, or 2. This solution is shown in Figure 54. Because real-valued filters are used, the resulting frequency responses are symmetric, so only the positive frequencies are shown in the figure. The scale values select the portions of the spectrum for the red, green, and blue images that are used to construct the output image.

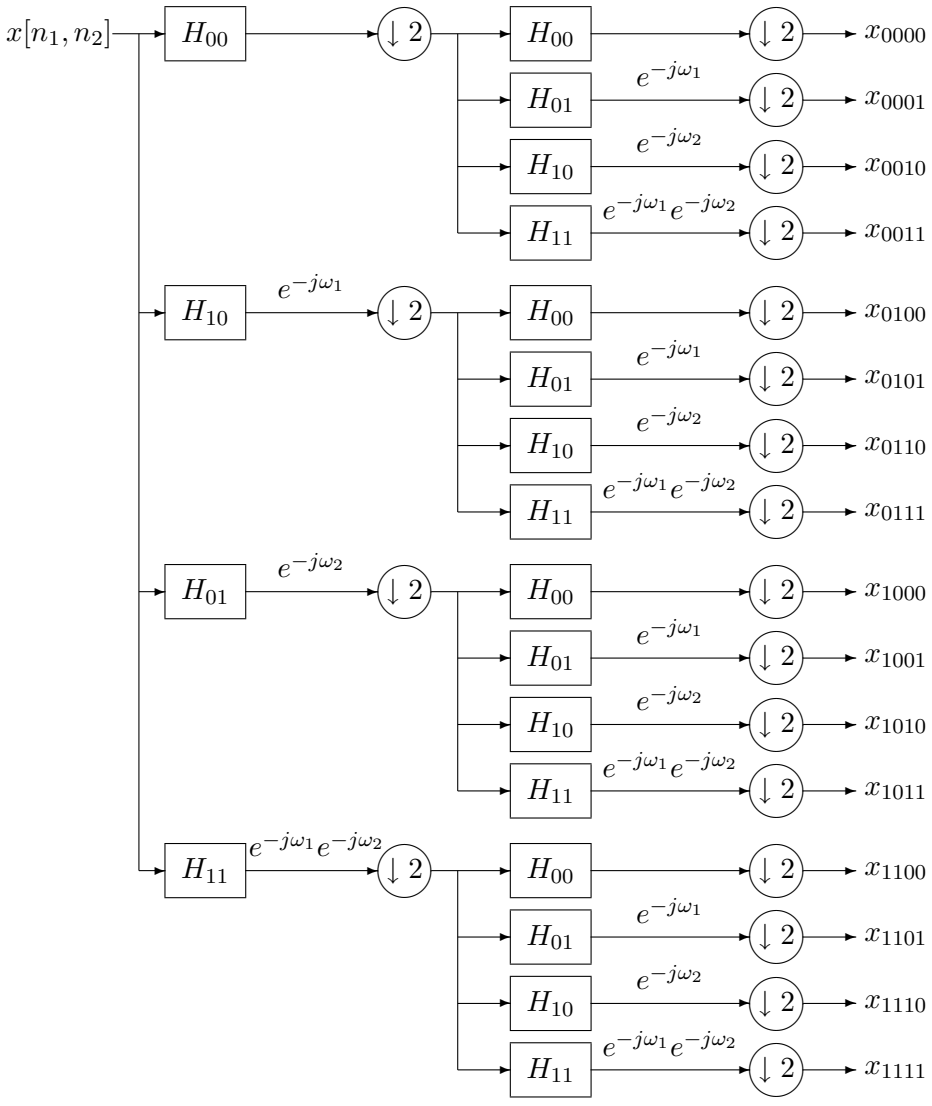


Figure 52: Tree-structured analysis bank block diagram.

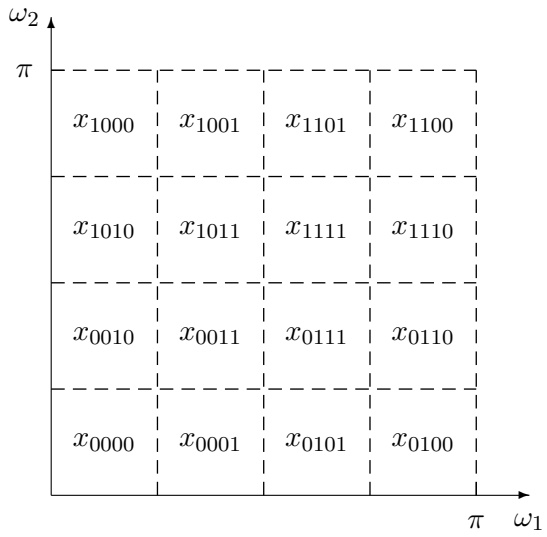


Figure 53: The three images are divided into frequency-selective regions of the spectrum, using a filterbank.

The red and blue subimages are used as correction terms to the green subimages. Therefore, the corresponding scaled subimages from the red, green, and blue images are added. The resulting set of subimages is passed through a synthesis filterbank, shown in Figure 55. In this figure, the inputs are the original subbands, $\mathbf{x}$. The subbands are scaled and combined by performing a dot product with the scale factor vector, $\mathbf{c}$. This scaled and combined set of subbands is applied to the input of the synthesis filterbank.

In Figure 53, the spectrum is divided into rectangular regions. This division allows us to use rectangular filters for the 2-D filterbank. This is an efficient choice since rectangular filters can be implemented using separable one-dimensional filters. Thus, a pair of one-dimensional analysis filters and a pair of one-dimensional synthesis filters are needed for the system. For a perfect reconstruction system, the synthesis filters can be described in terms of the analysis filters. Therefore, only one pair needs to be defined. The biorthogonal family of filters from the Matlab Wavelet Toolbox are good candidates for this application [24]. They are linear phase filters with exact reconstruction properties and can have short filter lengths so that ringing around edges is minimized [10]. The frequency response of an example set of analysis filters is shown in Figure 56. Given the analysis filters $h_0[n]$ and

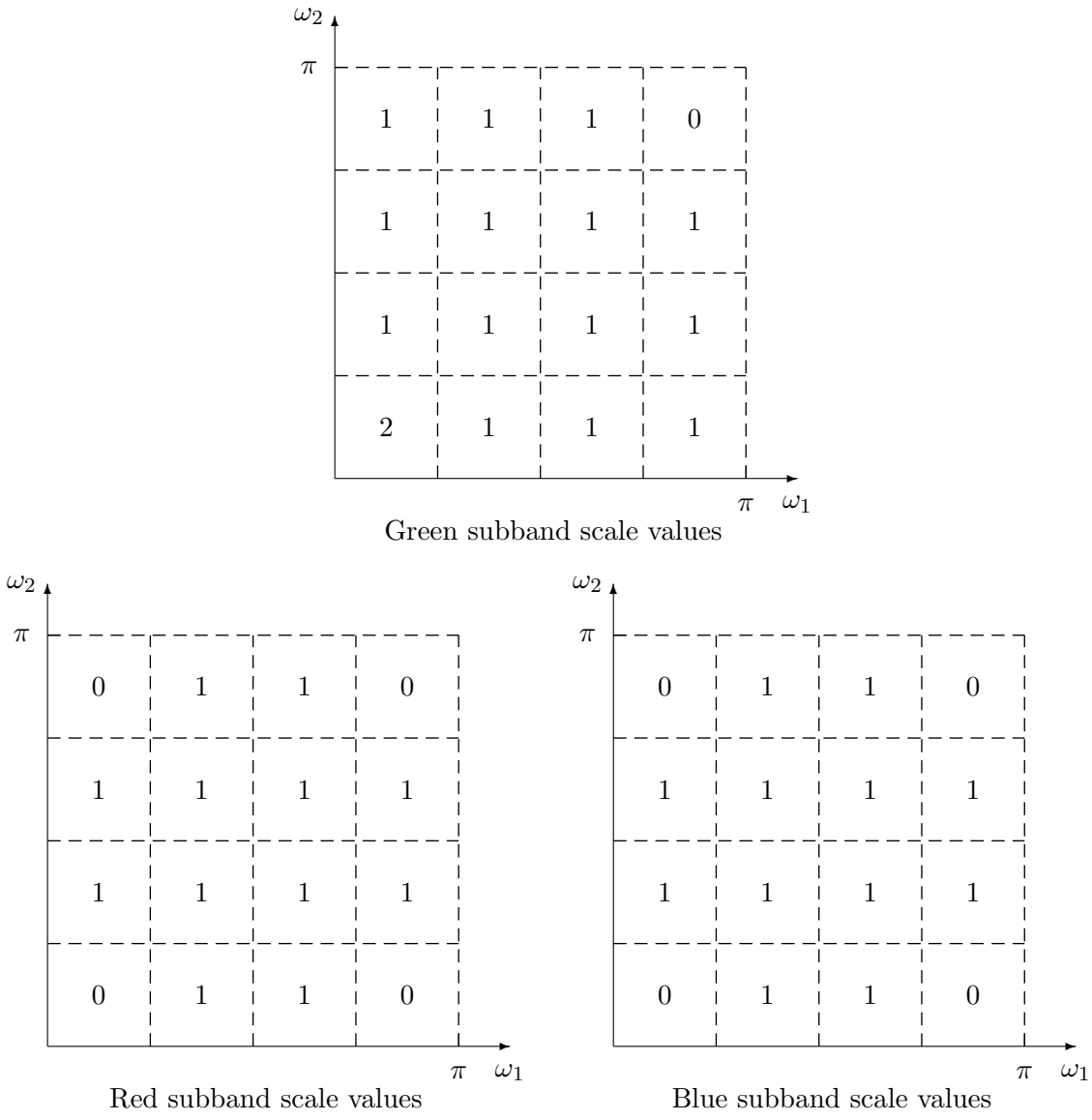


Figure 54: The subbands of the red, green, and blue image are scaled before combining and synthesizing the three images.

$$\mathbf{c} = \begin{bmatrix} c^r \\ c^g \\ c^b \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

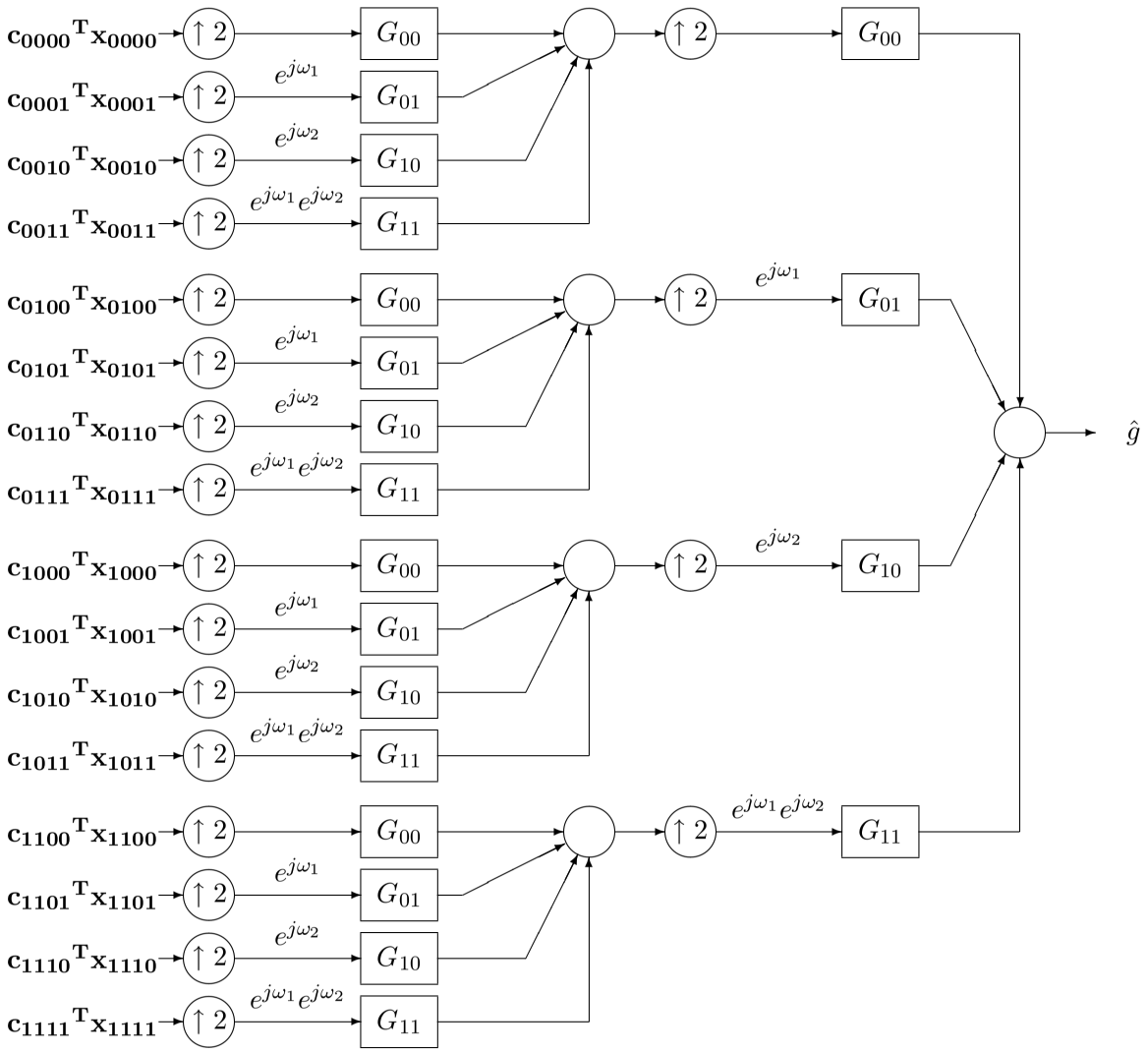


Figure 55: The subbands of the sampled image are scaled and then recombined using the synthesis filterbank.

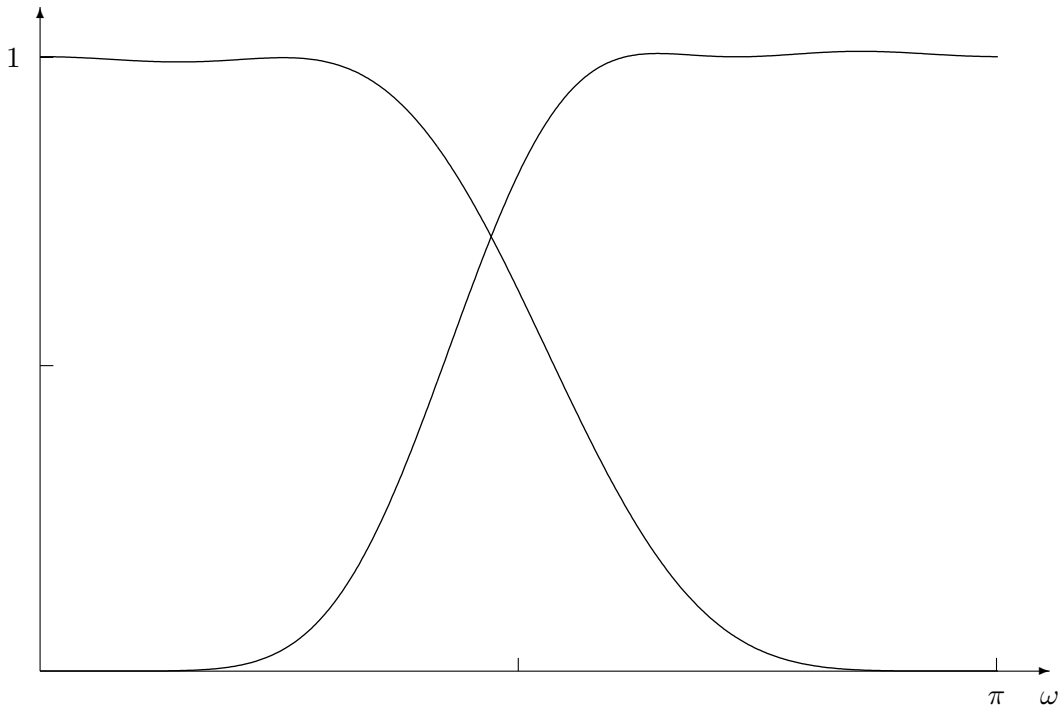


Figure 56: Matlab's biorthogonal wavelet filters, *bior6.8*.

$h_1[n]$, the reconstruction filters are determined by

$$\begin{aligned} g_0[n] &= -(-1)^n h_1[n] \\ g_1[n] &= -(-1)^n h_0[n]. \end{aligned} \tag{79}$$

The two dimensional filters are computed using

$$h_{00}[n_1, n_2] = h_0[n_1]\ h_0[n_2] \tag{80}$$

$$h_{01}[n_1, n_2] = h_0[n_1]\ h_1[n_2] \tag{81}$$

$$h_{10}[n_1, n_2] = h_1[n_1]\ h_0[n_2] \tag{82}$$

$$h_{11}[n_1, n_2] = h_1[n_1]\ h_1[n_2]. \tag{83}$$

The example analysis filterbank used a two-level decomposition. This provides sufficient frequency division so that the low-frequency portion of the spectrum is separated from the medium-frequency and higher-frequency portions. This allows the subbands to be scaled separately so that, for example, the r_{1111} subband containing a shifted copy of

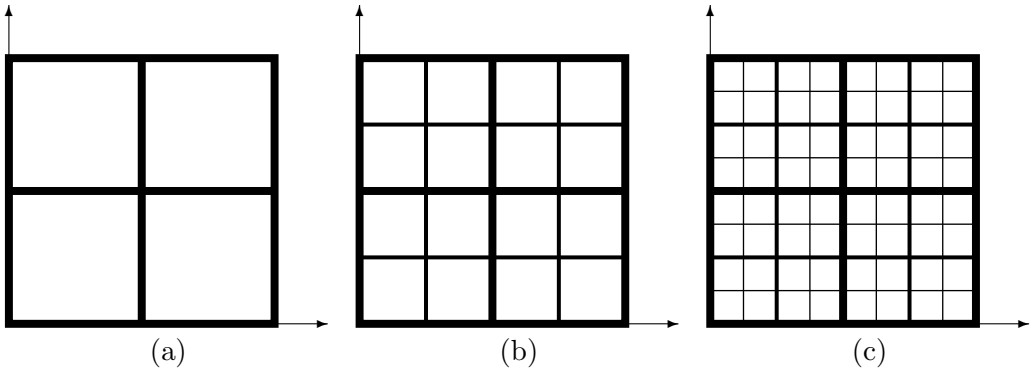


Figure 57: The spectrum divided by (a) one-level decomposition, (b) two-level decomposition, and (c) three-level decomposition.

the low-frequency spectrum can be blocked from the green output without blocking the mid-frequencies that aid in alias cancellation. Figure 57 shows the ideal subbands for a single-level, two-level, and three-level decomposition. In a one-level decomposition, no subbands isolate the medium frequencies, so one level is not sufficient. The three-level decomposition divides the subbands even further than the two-level decomposition. However, the area these subbands cover may be too small to provide added benefit. To show this, the difference image, $R-G$, was computed for each of the images in the *Kodak* color image database. An FFT of the difference image was computed. In the two-level decomposition, on average, 1.9% of the difference energy lies outside of the low-frequency subband. The highest percentage of energy from any one image is 6.5%. Whereas in the three-level decomposition, the average percentage outside the low-frequency subband is 6.5% and the maximum percentage is 26.9%. A simple illustration of this is shown in Figure 58. The gray area shows the highest concentration of energy in the difference image. This region covers most of the larger subband, thus dividing it further provides little benefit.

4.2.3 Results

Four sets of biorthogonal wavelet filters are used to evaluate this system. The filters used are the 'bior' family of filters from Matlab's Wavelet Toolbox. The responses of these filters are shown in Figure 59. Two of the filter sets have flat responses, *bior4.4* and *bior6.8*. Of these, *bior6.8* has the best subband separation with a short transition band and good stopband rejection.

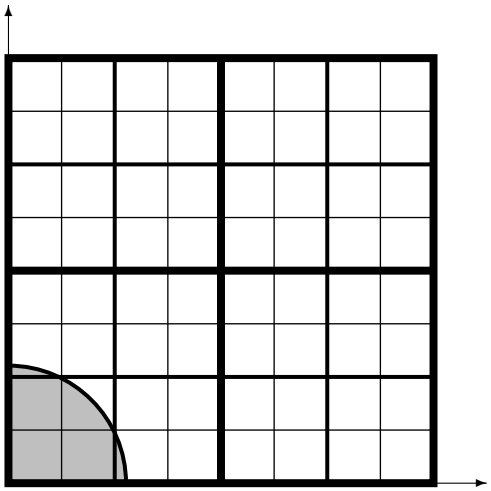


Figure 58: The gray region shows where most of the energy in a typical $R - G$ difference image is concentrated. This area mostly covers one subband of the two-level decomposition. Using a three-level decomposition, then, appears to have little benefit.

Figure 60 shows the chirp image result when interpolated using the four 'bior' filter sets. In all four result images, at the high horizontal and vertical frequencies, near $(0, \pi)$ and $(\pi, 0)$, the result is darker than the rest of the image. This shows that these regions are attenuated by this system. Only the *bior2.2* image in (a) shows further noticeable degradation. This region is highlighted in the figure. The reference images shown are from POCS and AH. The POCS image in (f) reconstructs the chirp signal best. The result of AH in (e) reconstructs the high frequencies along the horizontal and vertical axis very well, but fails to reconstruct features outside the diamond-shaped region. The comparison of the filterbank system to the reference algorithms show the improvements that can be made by using an edge-directed system like POCS or AH. Both edge-directed systems are able to reconstruct the high frequencies along the horizontal and vertical axis, providing sharper edges on horizontal and vertical features in the image.

The *Lighthouse* image was interpolated using the filterbank system with each of the sets of filters. Portions of the image are shown in Figure 61. Here, there's very little difference among the four filter sets. Yet, when the mean square error is computed for each image in the database, see Table 2, the *bior6.8* filter set consistently outperforms the other filter sets. Also, in Table 2, the mean square error is computed using the POCS algorithm. It

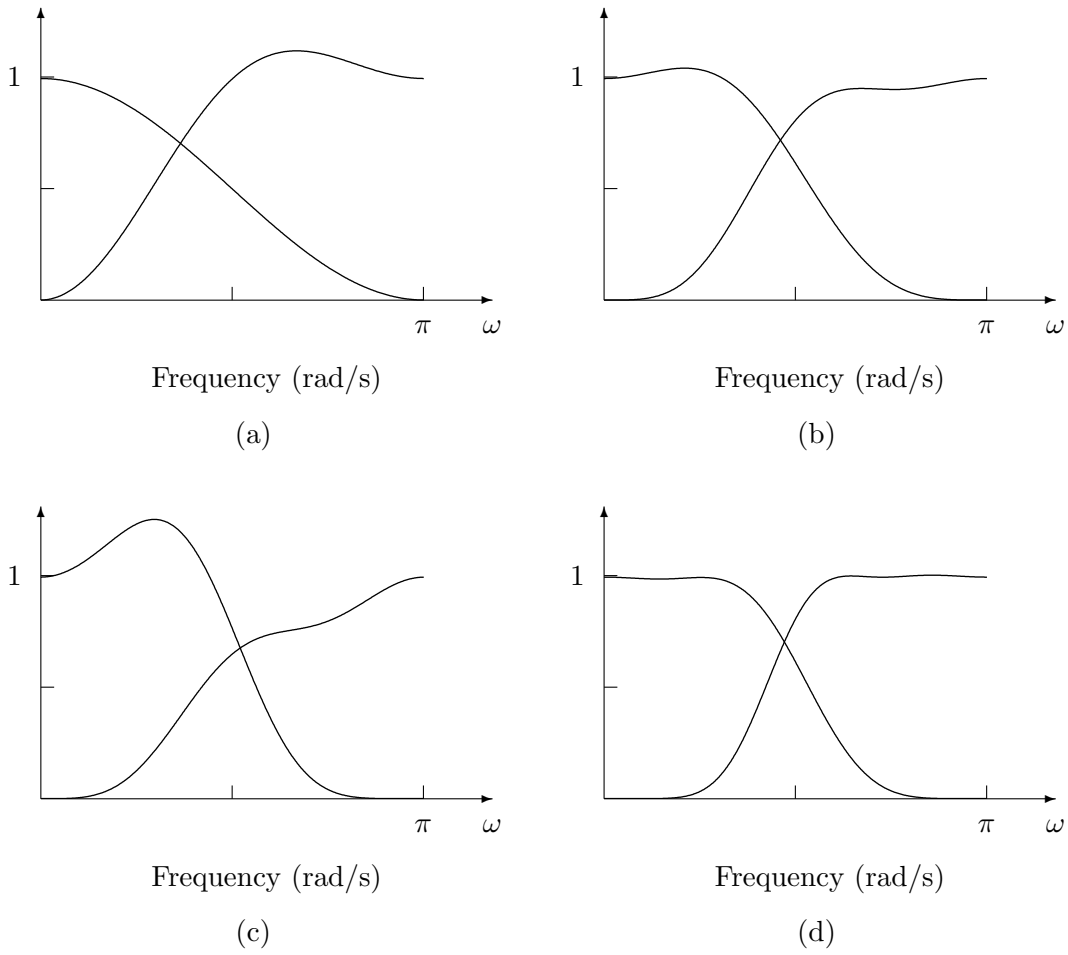


Figure 59: Matlab's biorthogonal wavelet filters: (a) *bior2.2*, (b) *bior4.4*, (c) *bior5.5*, (d) *bior6.8*.

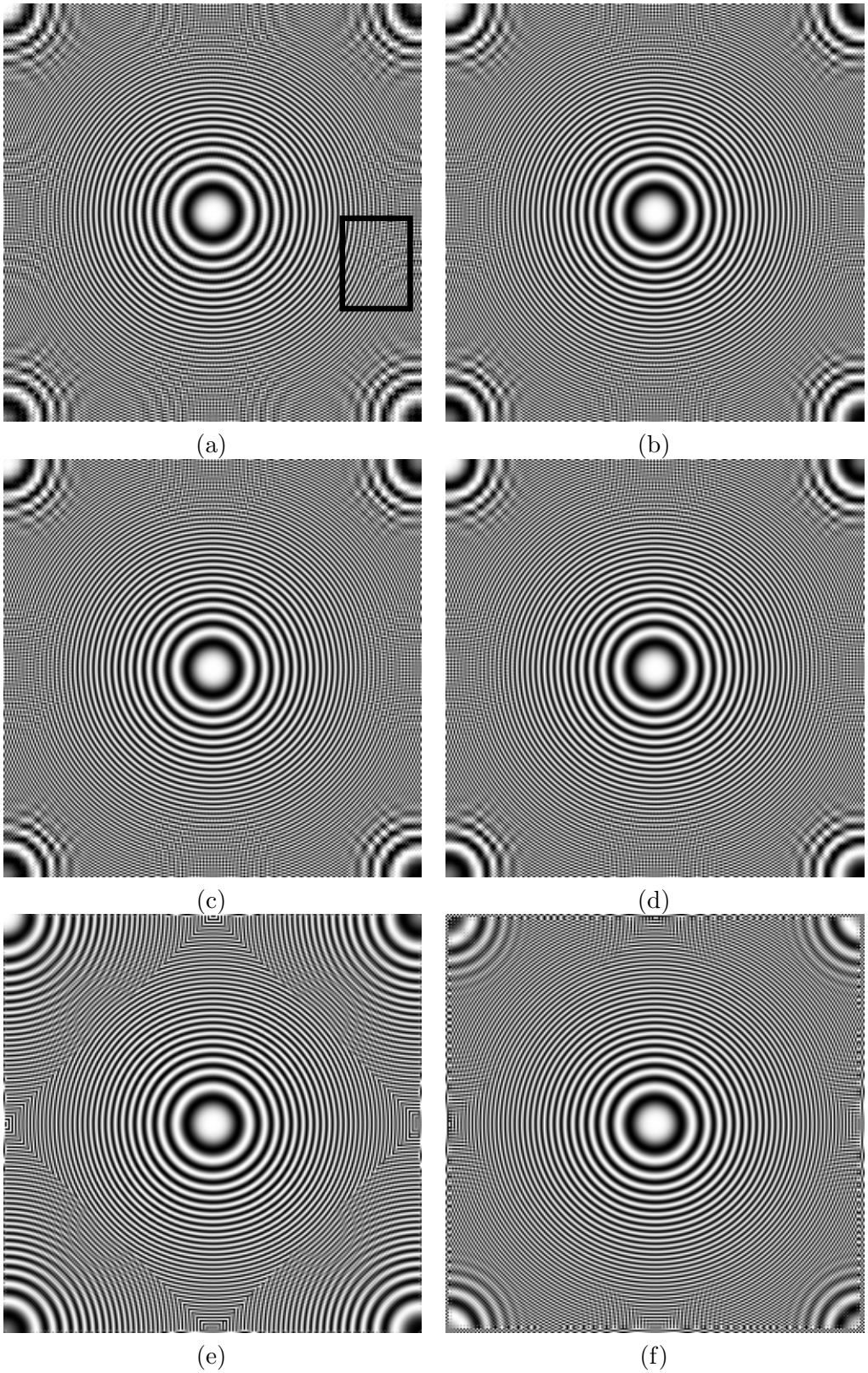


Figure 60: Chirp result images for green interpolation with filterbank system. (a) *bior2.2*. (b) *bior4.4*. (c) *bior5.5*. (d) *bior6.8*. (e) AH. (f) POCS.

Table 2: Mean square error results for each of 24 images in the Kodak color image database. Four different filter sets were used in the filterbank system and these results are compared to the POCS algorithm for the green image.

Image	*POCS*	*bior2.2*	*bior4.4*	*bior5.5*	*bior6.8*
1	8.22367	15.9171	14.0693	14.3022	13.8265
2	4.89985	8.50881	6.58401	6.72132	6.15593
3	3.35158	5.78861	4.24007	4.31327	3.8598
4	4.24448	6.85331	5.05622	5.16778	4.617
5	9.44767	16.1322	11.1128	11.2803	10.057
6	6.54433	13.546	12.0177	12.1415	11.7325
7	2.71291	5.65261	3.63886	3.73918	3.13977
8	12.2949	31.6855	28.5842	29.0065	28.2728
9	3.07211	6.17064	4.89852	4.97262	4.64064
10	2.5271	4.22537	3.24128	3.26912	3.05628
11	5.30737	9.89069	8.23812	8.40837	7.93246
12	2.81896	5.92518	5.03904	5.10579	4.83344
13	19.3029	25.0937	22.3748	22.7692	22.0295
14	10.9923	18.8984	14.0711	14.3788	12.8508
15	4.47751	8.59176	7.11807	7.12806	6.85857
16	2.71726	5.68632	5.21426	5.34774	5.1765
17	3.84755	5.09122	4.33889	4.39951	4.21708
18	10.6523	13.5011	11.2527	11.4807	10.7983
19	4.46177	11.8423	10.4971	10.6584	10.3134
20	19.0199	9.74616	8.04329	8.04729	7.32456
21	5.97037	9.66644	8.24816	8.36668	8.05243
22	7.7588	11.8785	10.225	10.3803	9.86578
23	2.43282	4.60105	2.94317	2.99732	2.63121
24	18.4275	19.9977	17.9115	18.1358	17.5762

can be seen that in most cases, the POCS algorithm outperforms the filterbank algorithm.

4.3 Analysis

Both approaches presented in this chapter to interpolate the green image are built on frequency-domain ideas. The input to the system contains three images sampled on different sampling grids. Due to the correlation among the input images, the information in the red and the blue images can be used to improve the interpolation of the green image.

Without the help of the red and the blue images, the green sampled image is filtered with a diamond-shaped filter. This reconstructs the original image as long as the image has no components that lie outside the diamond-shaped Nyquist sampling region. If an image

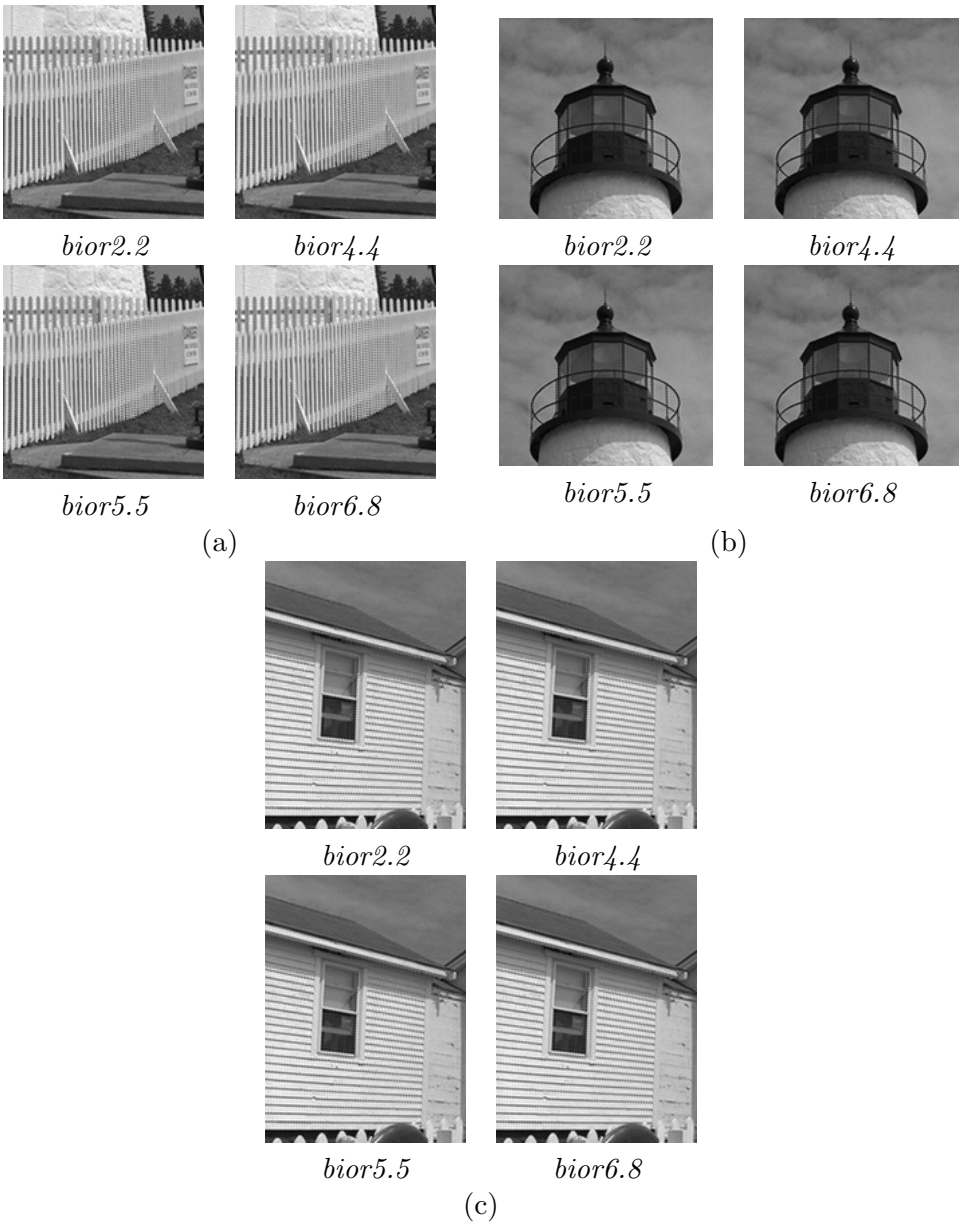


Figure 61: Portions of the *Lighthouse* green image interpolated with the filterbank system. Four pairs of biorthogonal wavelet filters are used. The results show little difference among the different filter sets.

contains these components, though, the diamond-shaped filter will pass aliased components to the output image, distorting the output.

One method of correcting this is to design a correction filter that is applied to the red and blue sampled images. This filter isolates mid-frequency components that are likely to be aliased in the green image. When this correction image is added to the interpolated green image, the aliased components are reduced. This method is modelled after the AH algorithm and the results showed that the mean square error for most images was lower for the general correction filter system than for AH.

A second method uses a filterbank system to reduce aliasing. This idea is based on the problem of determining how to combine the red, green, and blue images such that the green image is reconstructed perfectly throughout the spectrum and aliasing from the sampling of the original green image is completely cancelled. The solution to this problem is not practical because it requires using the red and blue images in parts of the spectrum where they are different from the green image. This includes the area around the origin in the spectrum. A filterbank system approximates the ideal solution by computing frequency-selective subbands. The subbands of the red, green, and blue images are scaled and combined. The combined set of subbands is passed through a synthesis filterbank to generate the output image. This approximate solution reduces aliasing in the output and improves edge response. This algorithm is based on the POCS method.

The filterbank system generally has lower MSE values than the correction filter system, but both systems have higher MSE values than the reference system described in the POCS demosaicking algorithm. The POCS algorithm uses edge-directed interpolation, which improves interpolation in the green image.

The next chapter introduces a new method to interpolate the red and blue images. The theory again concentrates on reducing aliasing in the output images.

CHAPTER V

COLOR FILTER ARRAY INTERPOLATION - RED AND BLUE IMAGES

The red and blue images are sampled on rectangular grids at half the rate of the green image. This results in a higher likelihood of aliasing in the red and blue images. Because of the phase difference between the red and blue sampled images, aliasing results in orange and blue distortions. These are very annoying in the output image. This chapter presents a method of interpolating the red and blue images that reduces the effects of aliasing.

In this system, information from the green image is used to cancel aliasing in the red and blue images. Again, it is assumed that high-frequency components are identical in all three images. Because the green image is sampled at a higher rate, the reconstructed green image contains most high-frequency components present in the original image. These can be isolated using highpass filters and modulated to approximate the aliasing in the red and blue images. With this estimate, the aliasing in the red and blue images is cancelled.

5.1 Background

Because the red and blue images are sampled on a rectangular grid, the Nyquist sampling limit is a rectangular region, as shown in Figure 62(a). This region contains half the spectral area of the diamond-shaped Nyquist region for the green image in Figure 62(b). Notably missing from the rectangular region are the horizontal and vertical high-frequency regions of the spectrum. This is highlighted in Figure 62(c). An ideal interpolation filter has the same shape as the Nyquist region and thus is unable to reconstruct high frequencies in the output image. This leads to blurring around edges in the image. Also, if the high frequencies are not blocked before the image is sampled, aliasing will be present in the sampled image and the ideal interpolation filter will not be able to distinguish aliased components from the low-frequency portion of the spectrum.

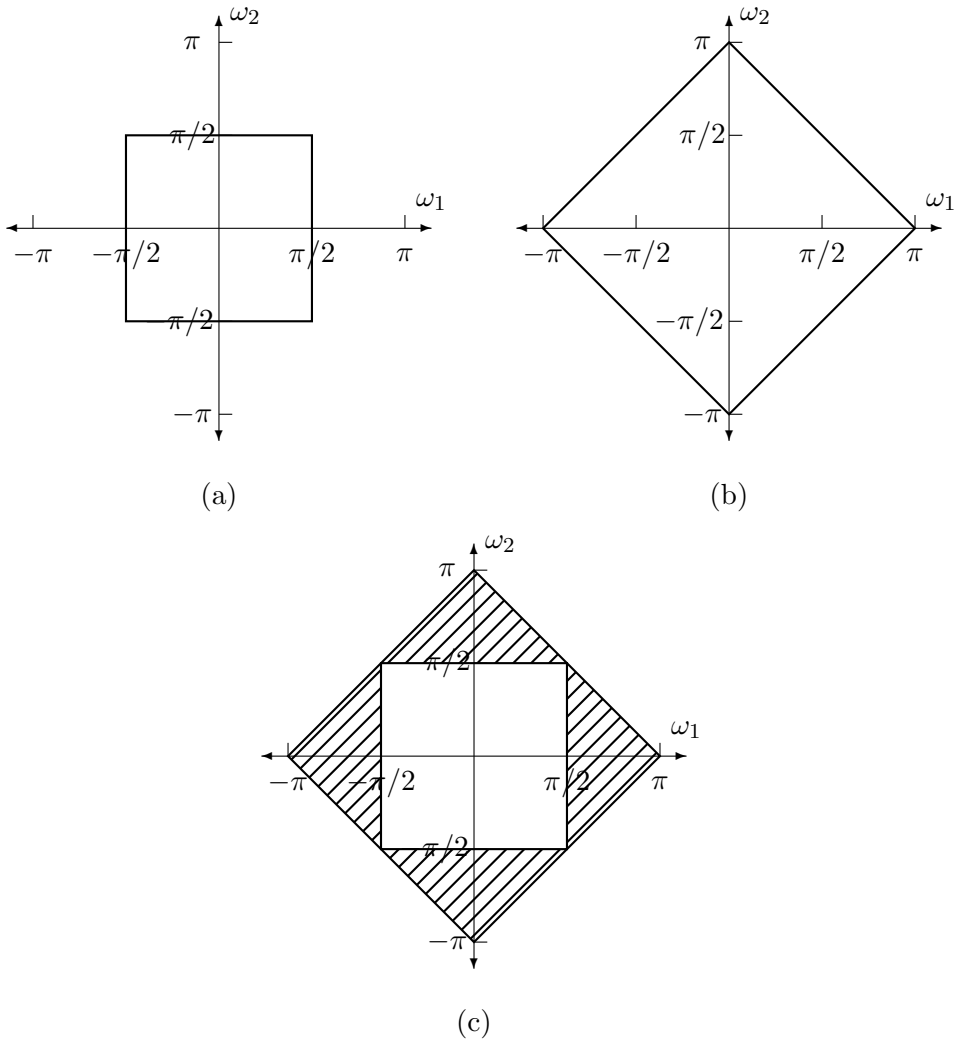


Figure 62: Nyquist regions for (a) rectangular sampling grid and (b) quincunx sampling grid. The rectangular sampling grid is missing the high horizontal and high vertical frequencies as shown in (c).

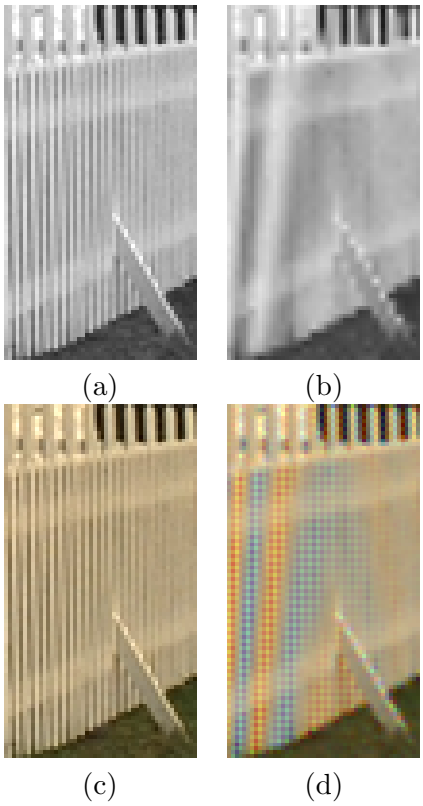


Figure 63: Example of aliasing due to insufficient rectangular sampling in the red and blue images. (a) Original red image. (b) Interpolated red image with aliasing. (c) Original color image. (d) Interpolated color image showing red and blue aliasing artifacts.

An example of this aliasing is shown in Figure 63. The original red image in (a) is sampled on a rectangular grid and interpolated with a rectangular interpolation filter. The resulting red image with apparent aliasing artifacts is shown in (b). In the color image, with both the red and blue images sampled on rectangular grids, these artifacts become more apparent. The original color image is shown in (c) and the result image in (d) is generated by interpolating with bilinear interpolation. The aliasing in the red and blue images have different phase and thus lead to objectionable color striping patterns.

The green image is sampled at twice the rate of the red and blue images. The difference between the two Nyquist regions is in the horizontal and vertical high-frequency regions. Since high-frequency components in the green sampled image are not aliased, they can be isolated using a highpass filter. By shifting the highpass filter output into the low-frequency region, the filter output can estimate the aliasing present in the red and blue images.

This is illustrated with an example in Figure 64. Suppose that an image contains a feature that generates a high-frequency component along the horizontal axis, as in Figure 64(a). Using the quincunx sampling grid of the green image, the spectrum is replicated once with a horizontal and vertical shift as in Figure 64(b). The rectangular sampling grid used for the red and blue images replicates the spectrum three times by shifting it horizontally, vertically, and both horizontally and vertically. This is shown in Figure 64(c). It is seen that the high-frequency components are replicated in the low-frequency region. Thus, when a rectangular interpolation filter is applied to this spectrum, the aliased components are passed to the output. In Figure 64(d), a horizontal highpass filter is applied to the green spectrum, isolating the high-frequency components. Modulating the filter output by multiplying by $(-1)^{n_1}$ shifts the high-frequency components into the low-frequency region as shown in Figure 64(e). Assuming that the high-frequency components in the green image are identical to the high-frequency components in the red and blue images, this shifted version of the green high-frequency components can be used as an estimate of the aliasing in the red and blue images. The modulated output of the highpass filter is combined with the interpolated red image to provide the corrected image with aliasing removed, as shown in Figure 64(f). Another benefit of isolating the high-frequency components in the green image is that these components can be added to the red and blue images to improve the sharpness of the image.

5.2 Algorithm Details

The sampled red image is described by the equation,

$$\begin{aligned} R_s\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}R\left(e^{j\omega_1}, e^{j\omega_2}\right) - \frac{1}{4}R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) \\ &\quad + \frac{1}{4}R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) \end{aligned} \tag{84}$$

and

$$G_s\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}G\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{2}G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) \tag{85}$$

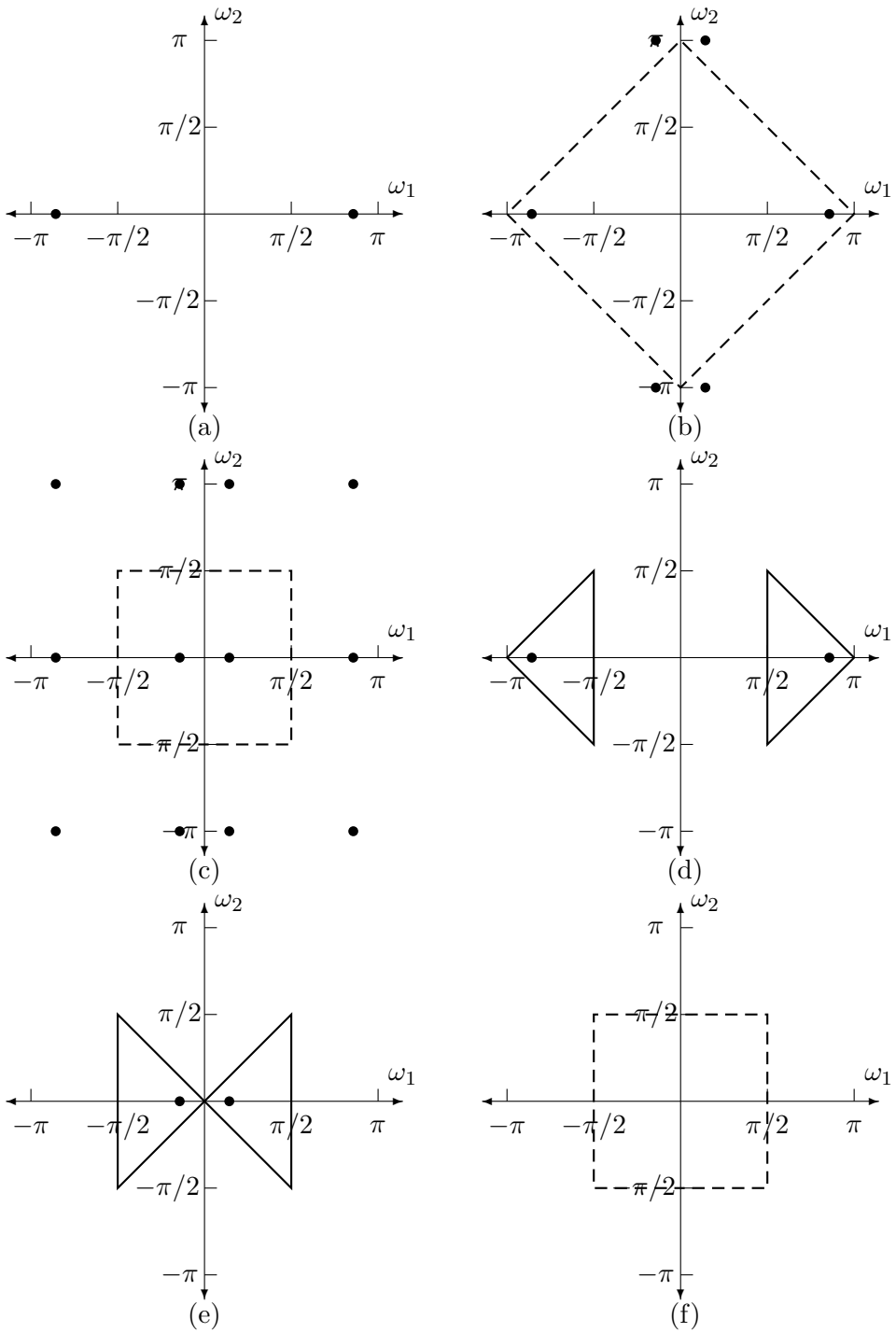


Figure 64: This example illustrates alias cancellation. (a) An image contains high-frequency components in the horizontal direction. (b) The original spectrum is shifted and replicated after sampling with a quincunx grid. (c) Sampling with a rectangular grid creates four copies of the original spectrum. The components inside the dashed region represent aliasing. (d) A horizontal highpass filter is applied to (b) to isolate the high-frequency components. (e) The output of (d) is shifted to the low-frequency region to estimate the aliasing in (c). (f) Applying a lowpass filter to (c) and subtracting (e) cancels aliasing in the output image.

describes the sampled green image. Applying a horizontal highpass filter to the sampled green image gives

$$G_h\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}\left[G\left(e^{j\omega_1}, e^{j\omega_2}\right) + G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right] H_g^h\left(e^{j\omega_1}, e^{j\omega_2}\right) \tag{86}$$

and modulating this horizontally gives

$$G_h^m\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}\left[G\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + G\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)\right] H_g^h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right). \tag{87}$$

The interpolated red image is given by

$$\begin{aligned} R_i\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}\left[R\left(e^{j\omega_1}, e^{j\omega_2}\right) - R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)\right. \\ &\quad \left. + R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right] H_r\left(e^{j\omega_1}, e^{j\omega_2}\right). \end{aligned} \tag{88}$$

The aliased components in the red image due to horizontal high frequencies are contained in the $R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right)$ term, and in the modulated green image, they are contained in the $G\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) H_g^h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)$ term. These terms when added, because of the opposite signs, cancel the horizontal aliasing in the interpolated red image.

We choose H_g^h to have the filter response shown in Figure 65(a). In Figure 65(b), the shifted copy of the diamond-shaped region is shown in the hashed region. The horizontal highpass filter is adjacent to the shifted copy of the vertical high frequencies. The filter passes horizontal high frequencies, while blocking the vertical high frequencies from the spectral copy, as shown in Figure 65(b), but if the horizontal highpass filter does not block the shifted copy of the vertical high frequencies, the second term in (87) adds to the third term in (88),

$$G\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) H_g^h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right),$$

to amplify the aliasing in the red image from the vertical high frequencies. Thus, it is important that the horizontal highpass filter block frequencies outside the diamond-shaped Nyquist region. If we make the assumption that the original image is bandlimited by the diamond-shaped Nyquist region, then the second term in (87),

$$G\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) H_g^h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right),$$

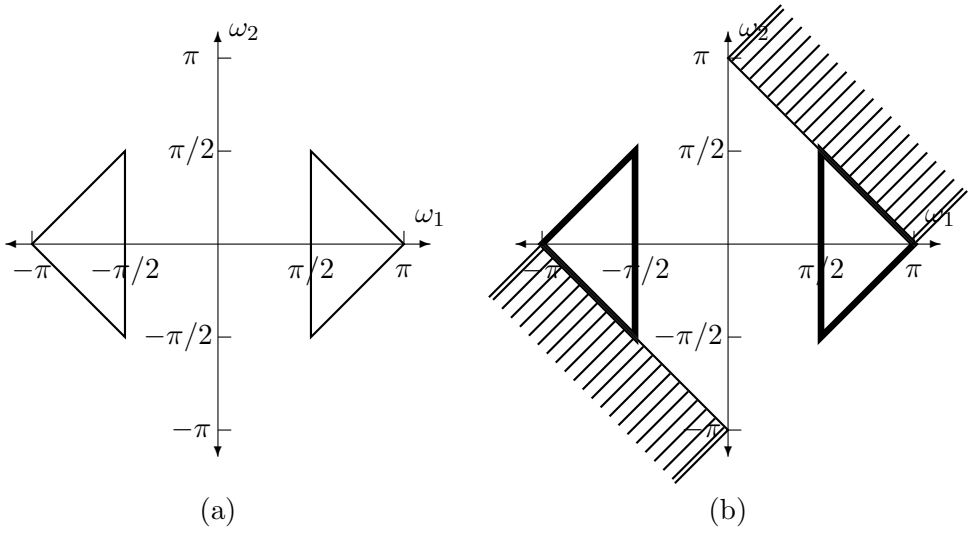


Figure 65: The horizontal highpass filter in (a) isolates high-frequency information from the green image. The filter must reject the vertical high-frequencies from the spectral copy of the original green spectrum. This transition is shown in (b).

contributes nothing to the output image because the energy from the shifted spectral copy lies completely outside of the passband of the horizontal highpass filter.

The horizontal highpass filter reduces the aliasing from the second term in (88), but a vertical highpass filter is also needed to cancel the vertical aliasing in the third term. This filter is applied in the same fashion to the green samples, as in

$$G_v\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}\left[G\left(e^{j\omega_1}, e^{j\omega_2}\right) + G\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right] H_g^v\left(e^{j\omega_1}, e^{j\omega_2}\right). \tag{89}$$

This result is modulated vertically to estimate the vertical aliasing in the red image,

$$G_v^m\left(e^{j\omega_1}, e^{j\omega_2}\right) = \frac{1}{2}\left[G\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) + G\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)\right] H_g^v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right). \tag{90}$$

Assuming that the high-frequency information is identical in the red and green images, the two terms $G\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) H_g^v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$ in (90) and $R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) H_r\left(e^{j\omega_1}, e^{j\omega_2}\right)$ in (88) combine to cancel each other. In the case of the horizontal aliasing, the two terms have opposite signs, so the terms are added. In the vertical aliasing case, the two terms have the same sign, so these terms are combined by subtracting the modulated green signal from the interpolated red image.

This algorithm obtains high-frequency information from the green image to cancel aliasing in the red and blue images. A highpass filter isolates horizontal high-frequency information and another highpass filter isolates vertical high-frequency information. The horizontal high-frequency information is shifted horizontally by π rad/sample to estimate the aliasing due to horizontal components in the red image and the vertical high-frequency information is shifted vertically by π rad/sample to estimate any aliasing from vertical components. The same procedure applies when interpolating the blue image. The only difference is how the aliasing estimate is combined with the interpolated blue image. The sampled blue image is described by

$$\begin{aligned} B_s\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}B\left(e^{j\omega_1}, e^{j\omega_2}\right) + \frac{1}{4}B\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) \\ &\quad -\frac{1}{4}B\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - \frac{1}{4}B\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right). \end{aligned} \tag{91}$$

The rectangular interpolation filter used for the red image, H_r, is also applied to the blue image, so

$$\begin{aligned} B_i\left(e^{j\omega_1}, e^{j\omega_2}\right) &= \frac{1}{4}\left[B\left(e^{j\omega_1}, e^{j\omega_2}\right) + B\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)\right. \\ &\quad \left. -B\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - B\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)\right] H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) \end{aligned} \tag{92}$$

describes the interpolated blue image. Therefore, the horizontal aliasing estimate should be subtracted from the interpolated blue image and the vertical aliasing estimate should be added to the interpolated blue image. This is opposite of how the aliasing estimates are combined in the red image. The phases of the $\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)$ term and the $\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$ term are opposite in the red and blue images because of the offset between the original red and blue sampling grids.

Besides cancelling aliasing in the interpolated red and blue images, high-frequency information taken from the green highpass filter outputs can be added to the interpolated red and blue images to improve the sharpness in the output images. The final red image is given by

$$\begin{aligned} \hat{R}\left(e^{j\omega_1}, e^{j\omega_2}\right) &= R_i\left(e^{j\omega_1}, e^{j\omega_2}\right) + G_h\left(e^{j\omega_1}, e^{j\omega_2}\right) + G_h^m\left(e^{j\omega_1}, e^{j\omega_2}\right) + \\ &\quad G_v\left(e^{j\omega_1}, e^{j\omega_2}\right) - G_v^m\left(e^{j\omega_1}, e^{j\omega_2}\right). \end{aligned} \tag{93}$$

The G_h and G_v terms add high-frequency information to the interpolated red image and the G_h^m and G_v^m terms reduce aliasing. The final blue image is described by

$$\begin{aligned}\hat{B}\left(e^{j\omega_1}, e^{j\omega_2}\right) &= B_i\left(e^{j\omega_1}, e^{j\omega_2}\right) + G_h\left(e^{j\omega_1}, e^{j\omega_2}\right) - G_h^m\left(e^{j\omega_1}, e^{j\omega_2}\right) + \\ & \quad G_v\left(e^{j\omega_1}, e^{j\omega_2}\right) + G_v^m\left(e^{j\omega_1}, e^{j\omega_2}\right). \end{aligned} \tag{94}$$

5.3 Implementation

This system reduces to a simple algorithm. In (93), the expression

$$G_h\left(e^{j\omega_1}, e^{j\omega_2}\right) + G_h^m\left(e^{j\omega_1}, e^{j\omega_2}\right) \tag{95}$$

is equivalent to

$$G_h\left(e^{j\omega_1}, e^{j\omega_2}\right) + G_h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) \tag{96}$$

since

$$G_h^m\left(e^{j\omega_1}, e^{j\omega_2}\right) = G_h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right). \tag{97}$$

In the time domain, this expression transforms to

$$\begin{aligned} & g_h[n_1, n_2] + (-1)^{n_1} g_h[n_1, n_2] \\ = \;& g_h[n_1, n_2]\left[1 + (-1)^{n_1}\right]. \end{aligned} \tag{98}$$

When n_1 is even, $[1 + (-1)^{n_1}] = 2$ and when n_1 is odd, $[1 + (-1)^{n_1}] = 0$. Therefore, this term is only non-zero when n_1 is even, as shown in Figure 66(a). For the vertical terms, because

$$G_v^m\left(e^{j\omega_1}, e^{j\omega_2}\right) = G_v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) \tag{99}$$

the expression

$$G_v\left(e^{j\omega_1}, e^{j\omega_2}\right) - G_v^m\left(e^{j\omega_1}, e^{j\omega_2}\right) \tag{100}$$

is equivalent to

$$G_v\left(e^{j\omega_1}, e^{j\omega_2}\right) - G_v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right). \tag{101}$$

The time domain representation for this is given by

$$\begin{aligned} & g_v[n_1, n_2] - (-1)^{n_2} g_v[n_1, n_2] \\ = \;& g_v[n_1, n_2]\left[1 - (-1)^{n_2}\right]. \end{aligned} \tag{102}$$

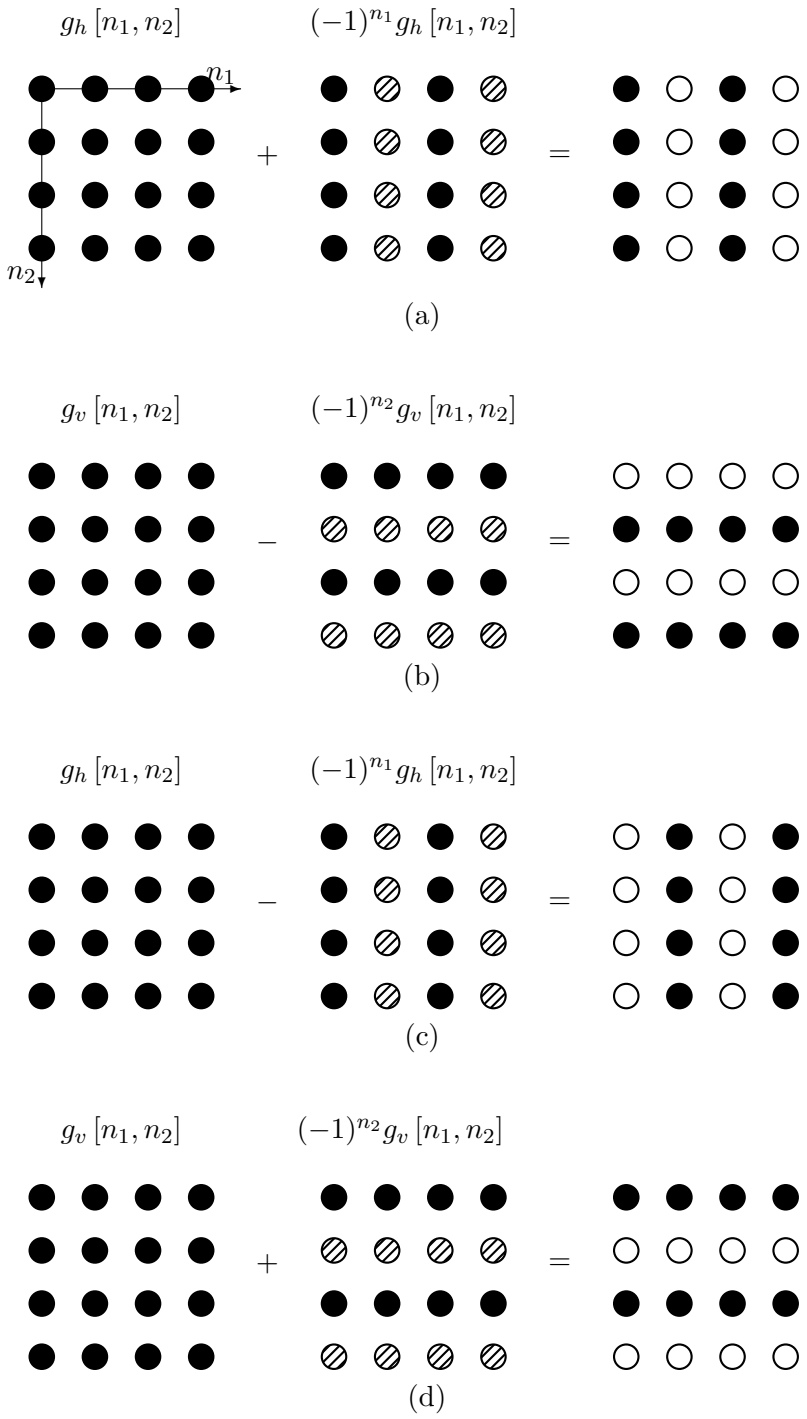


Figure 66: The interpolation algorithm for the red and blue images simplifies by combining the green correction terms with a modulated version. This generates entire rows or columns of zeros. (Solid black circles have no phase shift, sign is positive. Hashed circles have a 180° phase shift, sign is negative. In the output, solid circles are positive, but have a coefficient of 2 and empty circles are zero.)

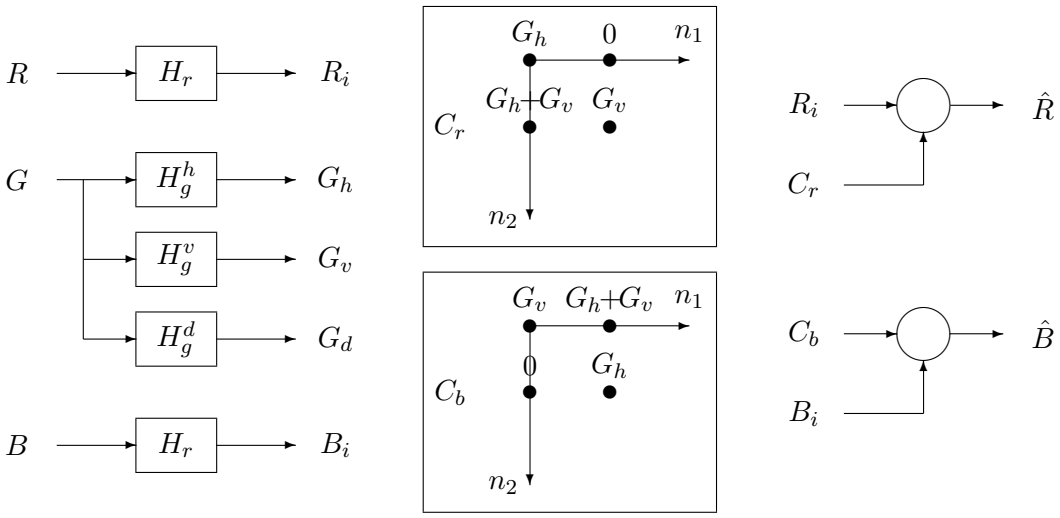


Figure 67: The red and blue images are interpolated and then corrected with high-frequency information from the green image.

Since $[1-(-1)^{n_2}]$ is non-zero only when n_2 is odd, the vertical terms only contribute on odd rows as shown in Figure 66(b). Therefore, the horizontal filter output is added on even columns and the vertical filter output is added on odd rows. At pixels located in an even column and an odd row (i.e. blue pixels), the correction is the sum of the vertical filter output and the horizontal filter output.

The same analysis applies to the blue image. In this case, the horizontal terms are non-zero only on odd columns and the vertical terms are non-zero only on even rows. Figure 67 illustrates the implementation of this algorithm.

5.4 Filter Design

To implement this algorithm, a rectangular interpolation filter is needed for the red and blue images. A highpass filter to capture the high frequencies in the green image is also needed. In the analysis, this highpass filter is divided between a horizontal highpass filter and a vertical highpass filter. By rotating the horizontal highpass filter kernel, we obtain a vertical highpass filter; therefore, only one highpass filter needs to be designed. The ideal filters shown in Figure 68 are used as models.

Two-dimensional rectangular filters are simple to design since they can be implemented as a separable filter, a one-dimensional filter applied to the rows of an image and then

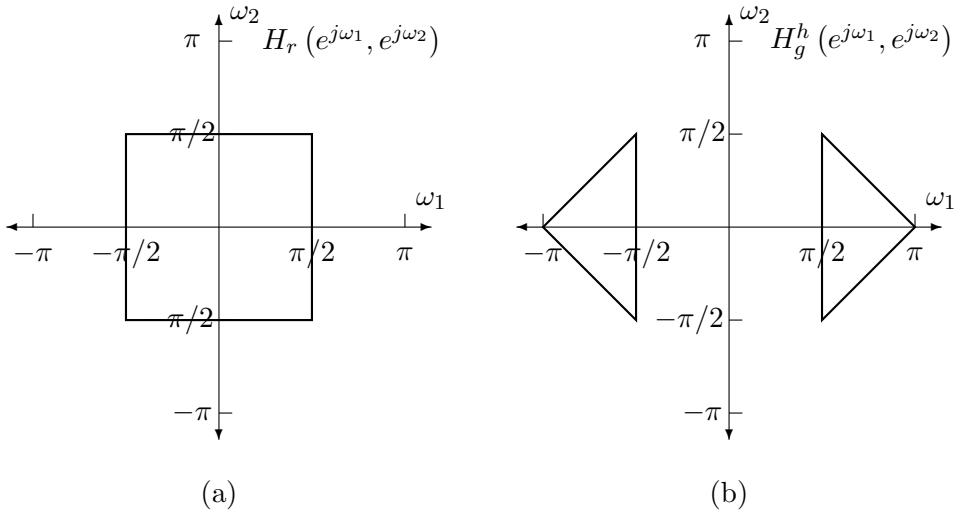


Figure 68: Ideal filters for the red and blue image interpolation algorithm. (a) Red and blue rectangular interpolation filter. (b) Green horizontal highpass filter.

applied to the columns of the result. Therefore, standard one-dimensional filter design practices are used to obtain the interpolation filter.

The highpass filter is not separable. An optimization routine is used to design this filter to match the ideal filter in some optimal sense. Two criteria are used for the optimization. In the frequency domain, the total squared error between the actual filter magnitude response and the ideal filter magnitude response is minimized. A second option is to minimize the maximum error between the actual and ideal filter magnitude responses.

For both cases, a transform is used to compute the frequency response at each point in a dense grid of the spectrum. Since these filters are symmetric, it is not necessary to consider the entire spectrum, but only the region, $(\omega_1, \omega_2) \in (0, \pi) \times (0, \pi)$. The 2-D discrete-time Fourier transform is given by

$$H\left(e^{j\omega_1}, e^{j\omega_2}\right) = \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} h[n_1, n_2] e^{-j\omega_1 n_1} e^{-j\omega_2 n_2} \tag{103}$$

where $h[n_1, n_2]$ is defined over $[-N_1, N_1] \times [-N_2, N_2]$. Because the filter is symmetric, $h[n_1, n_2] = h[-n_1, n_2] = h[n_1, -n_2] = h[-n_1, -n_2]$; thus, the transform can be restated as

$$H\left(e^{j\omega_1}, e^{j\omega_2}\right) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} h[n_1, n_2] c[n_1, n_2] \cos(\omega_1 n_1) \cos(\omega_2 n_2) \tag{104}$$

where

$$c[n_1, n_2] = \begin{cases} 1 & n_1 = 0 \text{ and } n_2 = 0 \\ 2 & n_1 = 0 \text{ or } n_2 = 0 \\ 4 & \text{otherwise} \end{cases} .$$

The transform is a linear operation so a transform matrix is now defined where each (ω_1, ω_2) point in the grid is converted to a row in our transform matrix, $\mathbf{W}$. Each column of $\mathbf{W}$ represents one of $[n_1, n_2] \in [0, N_1] \times [0, N_2]$. Each row is composed of $c[n_1, n_2] \cos(\omega_1 n_1) \cos(\omega_2 n_2)$. With this, the transform is converted to a matrix expression,

$$H = \mathbf{W}h \tag{105}$$

where h is a vector of coefficients whose coordinates align with the $[n_1, n_2]$ columns of $\mathbf{W}$. The desired filter response is also converted to a vector where each row represents the magnitude response at each point (ω_1, ω_2).

The total squared error is measured with

$$\xi^2 = (\mathbf{W}h - d)^T (\mathbf{W}h - d) \tag{106}$$

where d is the desired filter response. The solution is given by

$$h^* = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T d. \tag{107}$$

To minimize the maximum error in the spectrum, we extend h by one element, δ. This is used to provide a floor and a ceiling for the error between the actual filter response and the desired response. The augmented vector is given by

$$\bar{h} = \begin{bmatrix} h \\ \delta \end{bmatrix} . \tag{108}$$

The optimal solution is found using linear programming to solve the following problem [15, 23]:

$$\min_{h,\, \delta} \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} h \\ \delta \end{bmatrix} \quad \text{such that} \begin{bmatrix} -\delta \\ -\delta \\ \vdots \\ -\delta \end{bmatrix} \leq \mathbf{W}h - d \leq \begin{bmatrix} \delta \\ \delta \\ \vdots \\ \delta \end{bmatrix} . \tag{109}$$

If the rows of $\mathbf{W}$ are labelled such that

$$\mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix},$$

then for each row, two constraints are added to the linear programming problem:

$$\begin{aligned} w_i h - d &\leq \delta \\ -\delta &\leq w_i h - d \end{aligned}$$

which simplifies to

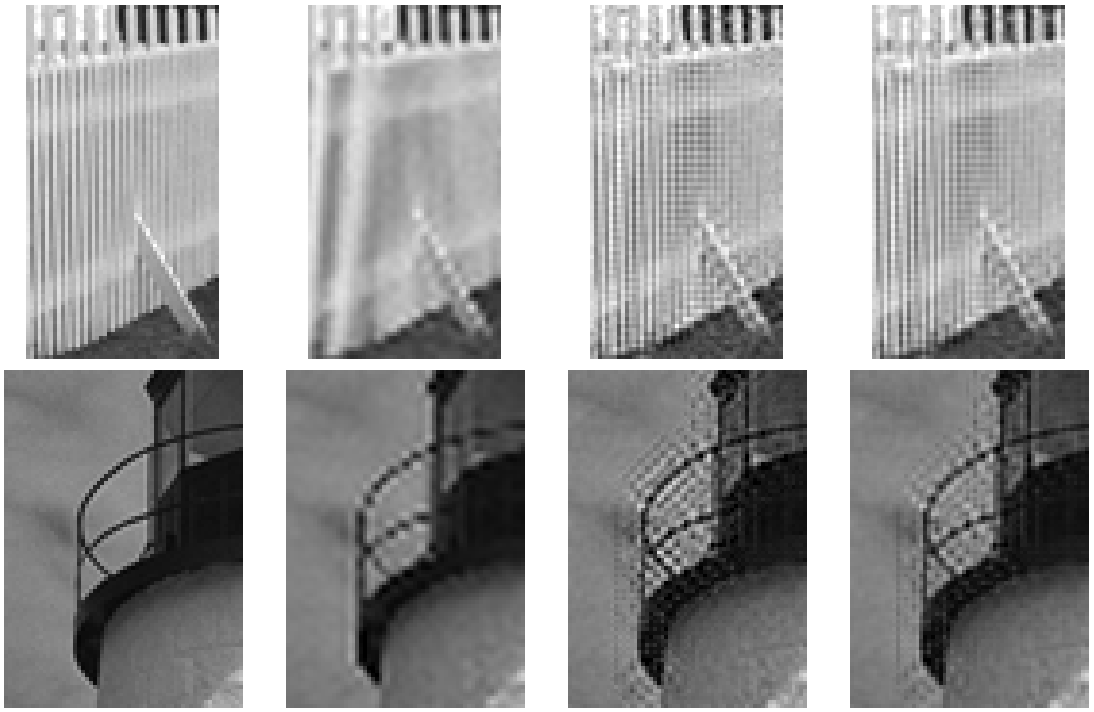
$$\begin{aligned} w_i h &\leq d + \delta \\ -w_i h &\leq -d + \delta \end{aligned}.$$

5.5 Initial Results

A prototype for a rectangular interpolation filter was designed by windowing an ideal sinc function with a Kaiser window. The least squares optimization in (107) was used to design a horizontal highpass filter for the green samples. As an alternative, another horizontal highpass filter was designed by minimizing the maximum error in the frequency domain, as in (109).

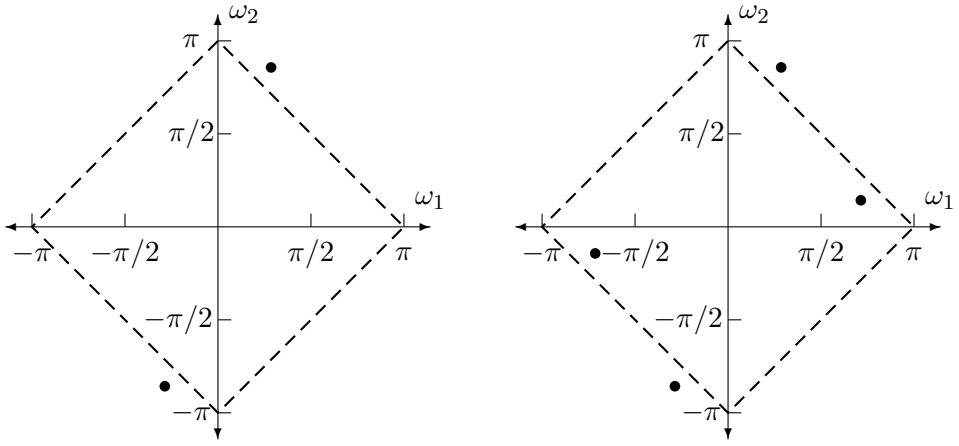
Both filters were placed in the interpolation system to cancel aliasing in the red and blue images. Portions of the *Lighthouse* red image are shown in Figure 69. The fence region shows that some alias cancellation has occurred. However, disturbing artifacts have now been introduced around the rails. These artifacts make this system unusable in a real camera system.

The artifacts result because of the failure of the assumption that all energy in the input image is contained within the diamond-shaped Nyquist region for the green image. When the green image has a component outside of the diamond-shaped region, yet primarily vertically oriented, as in Figure 70(a), it is confused as a horizontal component because the spectral copy replicates the component in the horizontal high-frequency region. This is shown in Figure 70(b).



(a) (b) (c) (d)

Figure 69: Portions of the red *Lighthouse* image are interpolated using a least squares filter and a minimax filter. (a) Original image. (b) Interpolated image using only the rectangular interpolation filter. (c) Interpolated result with optimal minimax filter applied to green image for correction. (d) Interpolated result with optimal least squares filter applied to green image for correction.



(a) (b)

Figure 70: The interpolation algorithm for the red and blue images assumed that all energy in the original image was contained in the diamond-shaped Nyquist region. (a) The original signal contains vertical high-frequency components outside the diamond-shaped Nyquist region. (b) After sampling, these high-frequency components are shifted in the horizontal high-frequency region and thus could be confused for horizontal components.

Instead of using the green samples to correct the red and blue image, the interpolated green image can also be used. If aliasing is removed in the interpolated green image, then this is a better source for correcting the red and blue images than the original green samples. In Chapter 4, we showed that the reference algorithm, POCS, provided the results with the least amount of error and minimized aliasing in the output green image. Therefore, the green image from POCS is used as the source image for correcting the red and blue images in the following two sections. This prevents aliasing in the green image from introducing artifacts in the red and blue images. The next section presents a simpler design technique for the red interpolation filter and the correction filters applied to the interpolated green image.

5.6 Improved Filter Design

In Chapter 3, the algorithm by Adams and Hamilton (AH) was described. (55), (59), (60), and (61) showed interesting properties for the filters used in this algorithm. The output equation for the red image was grouped by the four terms from the sampled red image: $R\left(e^{j\omega_1}, e^{j\omega_2}\right)$, $R\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right)$, $R\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right)$, and $R\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right)$. We showed that by combining the sampled red image with correction filters from the green image, the interpolation of the red image was improved. The effective filters are the combination of the correction filters and the interpolation filter that were factored from each of the four groupings. The effective filters had a simple kernel where the effective kernel for the $R\left(e^{j\omega_1}, e^{j\omega_2}\right)$ term is an identity filter and the others are zero filters.

The red interpolation in AH uses four filters: the red interpolation filter, a correction filter for the G_r pixels, a correction filter for the G_b pixels, and a correction filter for the B pixels. A general structure for the design of these filters is provided such that the set of filters exhibits the same properties as those used in AH.

From Chapter 3, the set of four filters must satisfy the following four properties.

$$\begin{aligned} H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_h\left(e^{j\omega_1}, e^{j\omega_2}\right) + & \\ H_v\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_d\left(e^{j\omega_1}, e^{j\omega_2}\right) & = 4 \\ -H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_h\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) - & \end{aligned} \tag{110}$$

Table 3: Description of five sets of coefficients to solve in filter design problem.

Case	Row Index	Column Index
1	0	0
2	Even	Even
3	Even	Odd
4	Odd	Even
5	Odd	Odd

$$H_v\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) + H_d\left(e^{j(\omega_1-\pi)}, e^{j\omega_2}\right) = 0 \tag{111}$$

$$H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_h\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - H_v\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) - H_d\left(e^{j\omega_1}, e^{j(\omega_2-\pi)}\right) = 0 \tag{112}$$

$$-H_r\left(e^{j\omega_1}, e^{j\omega_2}\right) + H_h\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) + H_v\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) - H_d\left(e^{j(\omega_1-\pi)}, e^{j(\omega_2-\pi)}\right) = 0 \tag{113}$$

Note that in (110), the 4 identifies a matrix where the central element is 4 and all other elements are zero. Each element in the matrices can be found independently by solving the set of four equations. The elements in the four filters are represented by the following variables:

$$\begin{aligned} a &: H_r \\ b &: H_h \\ c &: H_v \\ d &: H_d. \end{aligned}$$

The five sets of coefficients that need to be solved are described in Table 3. Case 1 solves the center element for each of the filter matrices. It is described by the set of system equations in (114).

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Longrightarrow \quad a = b = c = d = 1 \tag{114}$$

Case 2 solves for the other coefficients that have both even row and column indices. Case 1 handled the center element where the combined filter had a center element of 4 and case

2 handles the other coefficients where the combined filter has coefficients of 0. The system matrix is the same for both cases. The solution is shown in

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \Longrightarrow \qquad a = b = c = d = 0. \tag{115}$$

Up to now, the filters that satisfy the four constraints (for a 5x5 filter) are defined by

$$H_r = \begin{bmatrix} 0 & a_0 & 0 & a_1 & 0 \\ a_2 & a_3 & a_4 & a_5 & a_6 \\ 0 & a_7 & 1 & a_8 & 0 \\ a_9 & a_{10} & a_{11} & a_{12} & a_{13} \\ 0 & a_{14} & 0 & a_{15} & 0 \end{bmatrix} \quad H_h = \begin{bmatrix} 0 & b_0 & 0 & b_1 & 0 \\ b_2 & b_3 & b_4 & b_5 & b_6 \\ 0 & b_7 & 1 & b_8 & 0 \\ b_9 & b_{10} & b_{11} & b_{12} & b_{13} \\ 0 & b_{14} & 0 & b_{15} & 0 \end{bmatrix}$$

$$H_v = \begin{bmatrix} 0 & c_0 & 0 & c_1 & 0 \\ c_2 & c_3 & c_4 & c_5 & c_6 \\ 0 & c_7 & 1 & c_8 & 0 \\ c_9 & c_{10} & c_{11} & c_{12} & c_{13} \\ 0 & c_{14} & 0 & c_{15} & 0 \end{bmatrix} \quad H_d = \begin{bmatrix} 0 & d_0 & 0 & d_1 & 0 \\ d_2 & d_3 & d_4 & d_5 & d_6 \\ 0 & d_7 & 1 & d_8 & 0 \\ d_9 & d_{10} & d_{11} & d_{12} & d_{13} \\ 0 & d_{14} & 0 & d_{15} & 0 \end{bmatrix}. \tag{116}$$

The remaining coefficients are found by solving the other three cases. When the row index is even and the column index is odd, the coefficients satisfy the system of equations in (117).

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \Longrightarrow \qquad c = d = 0 \qquad b = -a \tag{117}$$

(118) gives the system of equations for Case 4 where the row index is odd and the column index is even.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \Longrightarrow \qquad b = d = 0 \qquad c = -a \tag{118}$$

The final case where the row index and the column index are both odd is described by (119).

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \Longrightarrow \qquad b = c = 0 \qquad d = -a \tag{119}$$

Substituting this information into our current filter solutions gives

$$H_r = \begin{bmatrix} 0 & a_0 & 0 & a_1 & 0 \\ a_2 & a_3 & a_4 & a_5 & a_6 \\ 0 & a_7 & 1 & a_8 & 0 \\ a_9 & a_{10} & a_{11} & a_{12} & a_{13} \\ 0 & a_{14} & 0 & a_{15} & 0 \end{bmatrix} \quad H_h = \begin{bmatrix} 0 & -a_0 & 0 & -a_1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -a_7 & 1 & -a_8 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -a_{14} & 0 & -a_{15} & 0 \end{bmatrix}$$
$$H_v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -a_2 & 0 & -a_4 & 0 & -a_6 \\ 0 & 0 & 1 & 0 & 0 \\ -a_9 & 0 & -a_{11} & 0 & -a_{13} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad H_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -a_3 & 0 & -a_5 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -a_{10} & 0 & -a_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{120}$$

This solution can be extended to larger filter sizes by following the same rules. The only design choice is the interpolation filter, H_r. Once this filter is defined, the other filters are completely determined.

5.7 Results

Using this simple design procedure, a 9x9 filter set was designed. Other sizes were also used, but the 9x9 size gave the best results. The correction filters are applied to the interpolated green image; therefore, the algorithm used to interpolate the green image must also be defined. POCS and the green filterbank system are used in this comparison. The filters from AH were also used with the green image from POCS. The red and blue interpolation results from POCS are presented for comparison.

Table 4: MSE Results for red and blue interpolation. (a) POCS. (b) AH-POCS. (c) Opt9-POCS. (d) Opt9-FB.

(a)		(b)		(c)		(d)	
Red	Blue	Red	Blue	Red	Blue	Red	Blue
20.62	15.04	11.06	12.47	11.85	13.43	32.90	31.52
9.26	10.21	9.39	8.83	8.90	8.84	27.24	17.89
5.41	7.88	4.41	7.65	4.22	7.04	7.68	9.47
8.74	8.14	8.44	6.29	7.69	6.53	15.56	8.93
12.81	18.10	10.56	20.17	10.28	18.92	19.88	22.86
11.07	18.19	8.16	13.11	8.85	13.51	26.07	29.57
4.83	8.01	4.02	8.45	3.77	7.52	10.32	12.62
24.33	31.37	17.00	23.47	18.03	24.30	70.13	72.82
4.78	7.44	3.55	7.04	3.63	6.59	9.79	10.87
6.59	9.44	3.68	6.47	3.68	6.35	8.09	9.35
9.35	9.80	8.16	9.44	8.22	9.68	19.62	17.61
5.73	9.80	3.59	5.19	3.60	5.12	11.57	12.07
27.07	37.43	21.33	32.61	23.91	34.36	42.90	52.29
16.83	25.55	16.63	26.66	14.82	24.98	27.13	25.65
13.58	13.30	10.27	8.89	9.81	8.96	23.74	15.50
6.14	8.00	3.83	5.57	4.16	5.71	13.09	13.86
6.55	10.84	3.76	7.60	3.97	7.69	7.82	10.16
15.74	20.91	11.48	22.16	11.90	22.03	19.45	24.58
9.31	9.93	5.92	10.21	6.27	10.28	23.45	26.50
13.85	10.78	4.41	11.71	4.63	11.54	13.54	19.82
10.29	13.31	7.64	12.63	8.24	12.91	17.88	21.56
10.85	15.03	10.13	15.76	10.18	15.30	16.60	20.71
3.91	6.49	4.89	7.66	4.57	7.17	7.49	6.09
23.75	38.63	19.84	36.90	20.55	37.61	25.72	43.13

Table 4 shows that when the AH filters are used with the green POCS image (AH-POCS), the results are better than the results from using the optimized filter set with the POCS green image (Opt9-POCS). In both cases, the red and blue images are interpolated with lower average error than the red and blue output images from POCS. As expected, because the filterbank system does not interpolate the green image as well as POCS, the green image from this system has less information to provide to correct aliasing in the red and blue images. Thus, these results (Opt9-FB) are significantly worse than the other methods.

Example results are shown in Figure 71. Each algorithm in Table 4 was used to generate the result from two portions of the *Lighthouse* red image, the fence region and the rail region. In the rail region, the images in (a)-(c) have little noticeable difference. According to this

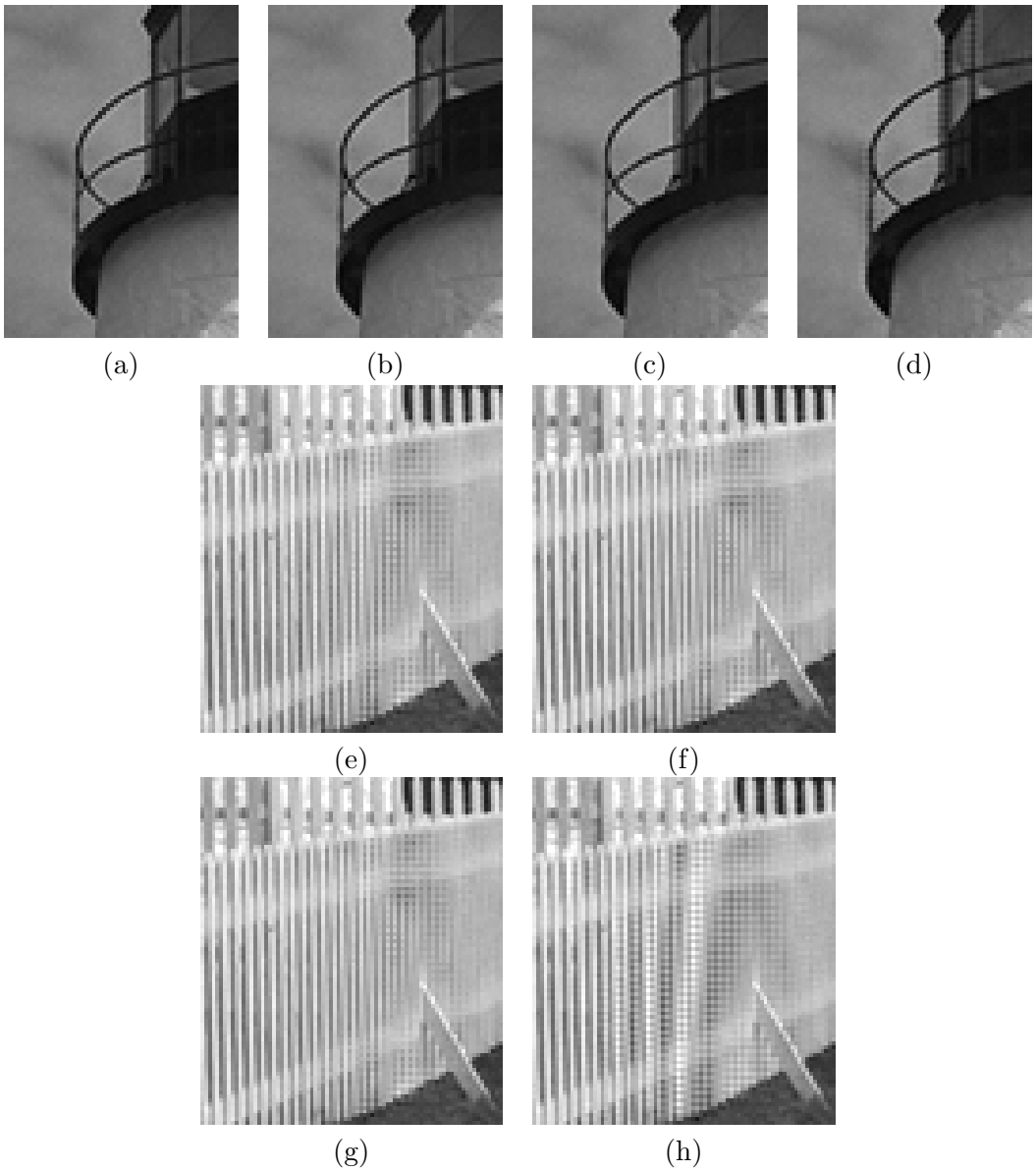


Figure 71: Fence and rail regions of red *Lighthouse* image interpolated by applying correction filters to an interpolated green image (interpolated by POCS or filterbank system). Two sets of correction filters were used, a 9x9 optimized filter set (Opt9) and the filter set defined by the Adams and Hamilton (AH) algorithm. Notation: The correction filter set is named first followed by the algorithm used on the green image, separated by a hyphen. (a) Rail: POCS. (b) Rail: Opt9-POCS. (c) Rail: AH-POCS. (d) Rail: Opt9-FB. (e) Fence: POCS. (f) Fence: Opt9-POCS. (g) Fence: AH-POCS. (h) Fence: Opt9-FB.

example then, the edge response of POCS and the correction algorithms are nearly the same. The image in (d) contains apparent artifacts around the rails. The fence region gives an example of how each algorithm reduces aliasing due to repeated patterns. The two images in (f) and (g) show the best response. The image in (e) was generated by the POCS algorithm and contains more aliasing than the images in (f) and (g). These images have the correction filters applied to the green image from POCS to correct the interpolated red image. This shows that this correction step gives better results with respect to aliasing than the POCS method. Thus, as expected, using the green image from the filterbank system is insufficient to completely remove aliasing from the interpolated red image. This result is expected because we showed that the filterbank system attenuated high-frequencies along the horizontal and vertical directions. Therefore, these components are not available in the output green image and cannot be used to completely estimate aliasing in the interpolated red image. Using edge-directed interpolation, both AH and POCS avoid these problems and thus, provide better results, not only in the green image, but also in the red and blue images.

Both the objective measures and the example images show that the correction filter system used by AH provides better results than the POCS system when the correction filters use a high-quality green image, such as that provided by the output of the POCS green interpolation algorithm.

5.8 Analysis

This chapter examines the problem of interpolating the red and blue images using the green image to significantly improve the interpolation, mainly by reducing aliasing in the output image. Because the green image is sampled at a higher rate than the red and blue images, the output green image is more likely to reproduce high-frequency components. These high frequencies are isolated from the green image and modulated to estimate the aliasing in the red and blue images.

We found that although applying the filters to the green samples directly does cancel aliasing in the output image, it also generates annoying artifacts around edges in the output

image that make this method unusable. However, if the interpolated green image is used as the high-frequency source, the results are quite good. Using the POCS algorithm to interpolate the green image and a set of alias cancelling filters, the red and blue images were interpolated with very small amounts of error and with results superior to the original POCS algorithm. The AH-POCS system is also a simpler algorithm to implement than POCS because it uses only 3x3 FIR correction filters.

Up to this point, the red, green, and blue results have been presented independently. Chapter 6 presents color results and provides a comparison to other algorithms from the literature.

CHAPTER VI

COMPARISON

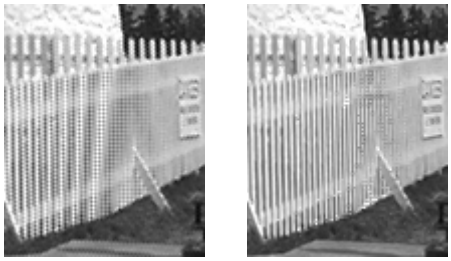
Two algorithms from the literature have been discussed in this thesis, the Adams and Hamilton edge-directed interpolation (AH) and POCS. These have been discussed because of their simplicity and also because they produce very good results. Other algorithms that combine red, green, and blue information are discussed in this section.

6.1 Other Algorithms from the Literature

In a patent [9], Cok described a method of interpolating the red and blue images. This patent assumes the green image has been interpolated, so every pixel has a green value. At the red and blue pixels, a ratio R/G or B/G is computed. The ratio images, R/G and B/G, are then interpolated to the full size of the grid. The patent description uses bilinear interpolation to interpolate the ratio values. The missing R and B values are computed by multiplying the interpolated ratio values by the green value at each pixel.

Experiments with this algorithm show that the quality of the interpolation depends upon the interpolation of the green image. Using AH (described in Section 3.1) to interpolate the green image, the red and blue images are interpolated well. However, when using bilinear interpolation for the green image, the red and blue images show significant aliasing artifacts. This algorithm can produce good results if it is paired with a good interpolation method for the green image. This can be seen in Figure 72. Here, the fence region from *Lighthouse* is shown. In Figure 72(a), the green image was interpolated with bilinear interpolation and then the red image was interpolated with hue-smoothing interpolation. The fence region contains a significant amount of aliasing. In Figure 72(b), the green image was interpolated with AH before the red image was interpolated with hue-smoothing. Much of the aliasing has been removed in this result.

Kimmel [17] also used a directional interpolation approach for this problem. Kimmel



(a) (b)

Figure 72: Hue-smoothing interpolation results [9]. (a) Red interpolated image after interpolating green image with bilinear interpolation. (b) Red interpolated image after interpolating green image with AH [13].

presents an image model where the color ratios G/R and G/B are constant within an object. The green image is interpolated to provide an initial estimate (any standard method is sufficient). Along four directions, weights are computed based on local gradients. If the change along a specific direction is large, the weight for that direction will be small. To compute the missing red and blue pixels, the color ratios R/G and B/G are interpolated using a weighted average. These color ratios are multiplied by the green values at these pixel locations to give the estimated red and blue values. The estimated green values are then updated using the directional weights and the ratios G/R and G/B. This process is iterated three times.

Kimmel's method combines the AH method with interpolation of hue values. The AH method uses only one direction while Kimmel's method allows the use of all directions with the weighted average.

Chang et al. [8] also presented a directional interpolation algorithm. This algorithm uses eight directions and the final output is interpolated from pixels in a combination of these directions. A diagram of the directions used is shown in Figure 73. An adaptive threshold is computed based on the local gradients for each of the directions. Directions with gradients less than the threshold are used in the interpolation. The missing color values are computed by adding the measured value at the pixel and the average color difference between the color being estimated and the measured color. The average color difference is computed along the chosen directions. The authors claim that using more than two directions helps to overcome areas where neither a horizontal interpolator nor a vertical

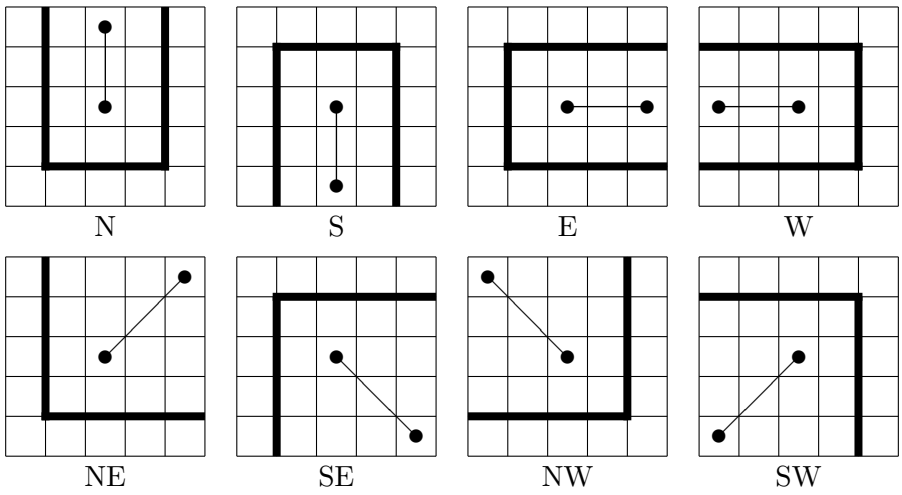


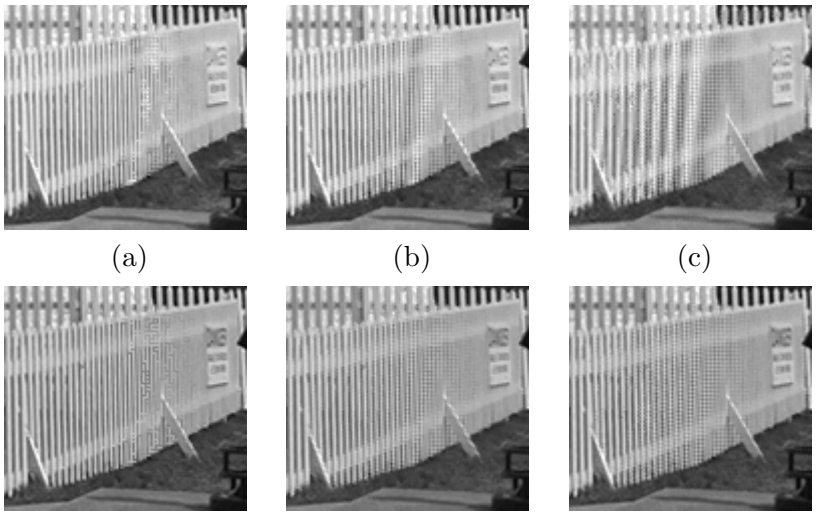
Figure 73: Directions used to measure gradients in Chang's variable gradient approach [8].

interpolator are appropriate. However, experiments show that results from this method are inferior to the results of AH.

Results from the three methods discussed can be seen in Figure 74 and Figure 75. These results show that AH performs best among the three. The addition of more directions, as in Chang's method, doesn't improve the interpolation result. Aliasing artifacts are more apparent along the fence in the image produced by Kimmel's method as compared to the result image of AH.

6.2 Objective Comparison

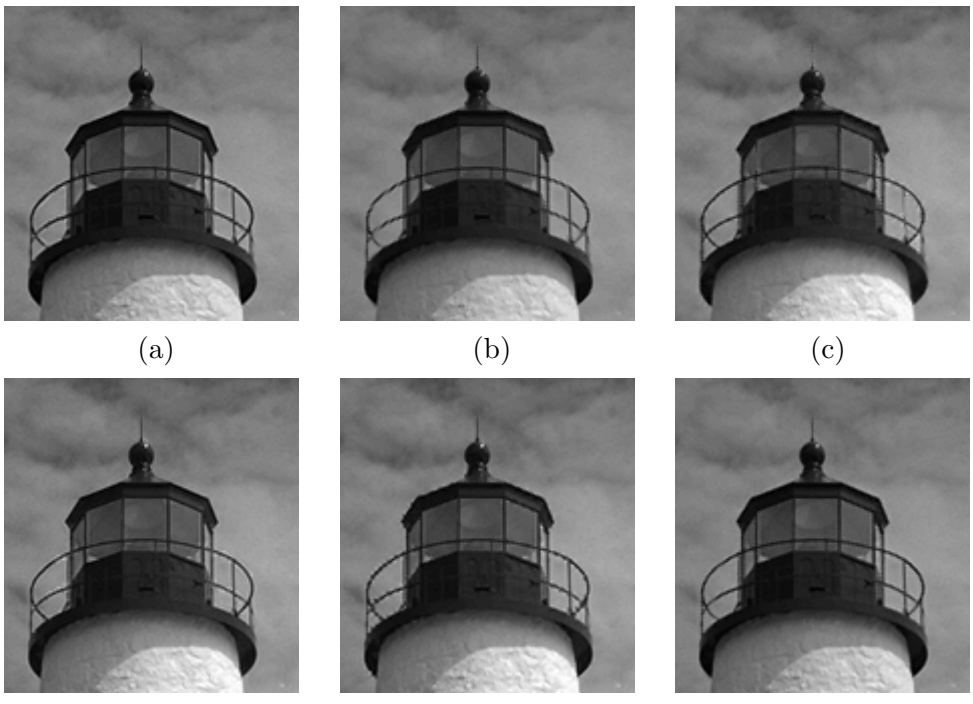
The 24 test images from the Kodak color image database are used to obtain objective results from eight algorithms, five from the literature (Kimmel, AH, Chang, POCS, and hue-smoothing interpolation), and three described in this thesis (Opt9-FB, Opt9-POCS, and AH-POCS). The last three have the correction filters applied to the interpolated green image as described in Section 5.6. The naming convention labels the red and blue correction filters first (Opt9 is a 9x9 optimized filter set and AH refers to the filter set defined by the Adams-Hamilton algorithm) followed by the algorithm used to interpolate the green image (FB refers to the filterbank system used on the green image). The results in Figure 76 show that the methods using the POCS algorithm for the green image perform best on average. Our example image in Figure 71 showed slightly better performance in the red and blue



(a) (b) (c)

(d) (e) (f)

Figure 74: Interpolation results from directional algorithms (fence image). (a) AH [13], red image. (b) Kimmel method [17], red image. (c) Variable gradient method [8], red image. (d) AH [13], green image. (e) Kimmel method [17], green image. (f) Variable gradient method [8], green image.



(a) (b) (c)

(d) (e) (f)

Figure 75: Interpolation results from directional algorithms (rail image). (a) AH [13], red image. (b) Kimmel method [17], red image. (c) Variable gradient method [8], red image. (d) AH [13], green image. (e) Kimmel method [17], green image. (f) Variable gradient method [8], green image.

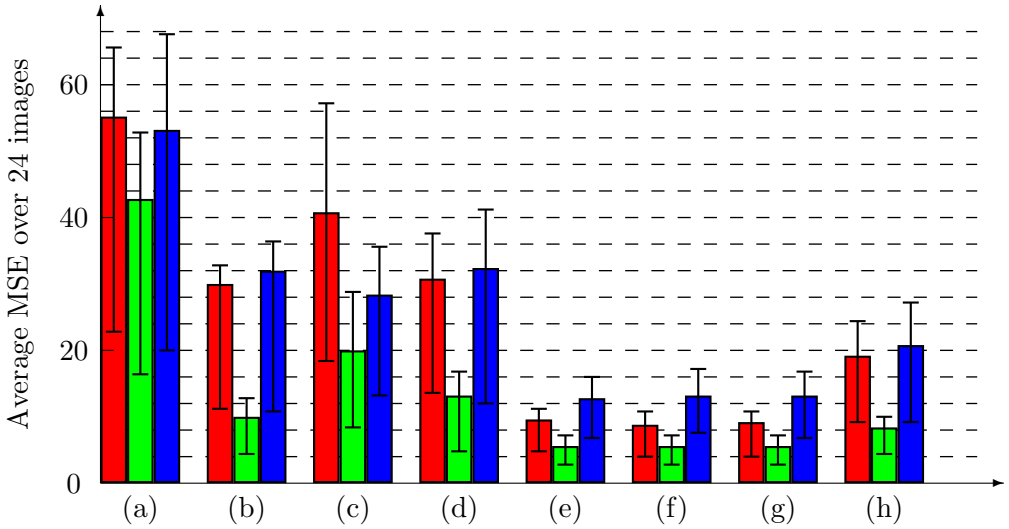


Figure 76: Average mean square error (MSE) over set of 24 color images. The error bars show the inter-quartile range (IQR). The MSE is reported for each of the red, green, and blue images independently, marked by the color of each bar. (a) Smooth-hue interpolation. (b) Chang's variable gradient method. (c) Kimmel's interpolation. (d) AH. (e) POCS. (f) AH-POCS. (g) Opt9-POCS. (h) Opt9-FB.

images by the AH-POCS and Opt9-POCS algorithms than the POCS algorithm. The graph confirms this for the red image, but error values for the blue image are slightly lower in the original POCS routine than the AH-POCS and Opt9-Pocs. Average error levels from the three algorithms are roughly the same, so any differences are very minor.

6.3 Color Image Results

Up to this point, results have been presented for each color channel independently. In this section, full color images are presented. For each example image, the original image is presented. This image is downsampled according to the Bayer CFA, and then interpolated with three algorithms. For comparison to show the problems with a simple algorithm, the images are interpolated with bilinear interpolation. The final two images show the result using POCS and the result using POCS on the green image and the Adams-Hamilton correction filters for the red and blue images (AH-POCS using notation in this thesis).

The first image contains a row of houses in an alpine village and contains many sources of aliasing as seen in the bilinear interpolation result in Figure 77(b). The wood trim and lattices around the houses provide sharp edges at various angles. These edges produce

significant amounts of aliasing. In Figure 78, the aliasing is greatly reduced in both result images. In (a), the POCS algorithm does a very good job interpolating the image. In (b), the simpler algorithm, AH-POCS, produces nearly the same result.

The second image is the *Lighthouse* image, a color version of the example used throughout this thesis. As seen in Figure 79, both the building and the fence are sources of aliasing. The bilinear interpolation image also appears less sharp than the original image. Figure 80 shows the interpolation result using POCS and AH-POCS. Along the fence, AH-POCS (b) removes more aliasing than POCS (a). Otherwise, the two images are very similar. Both significantly reduce aliasing and improve the sharpness of the image.

Shown in Figure 81, the third example is the *Motorcycles* image. Problem areas in this image include the spokes on the wheels. Again, these represent sharp edges at various angles. The bilinear interpolation result shows some aliasing around the spokes. In the result images in Figure 82, both POCS and AH-POCS remove most of the aliasing to produce very good images.

These results show that the reference algorithm POCS produces a significant improvement over bilinear interpolation. This algorithm includes an iterative process using filterbanks to interpolate the red and blue images. AH-POCS uses linear filters to interpolate the red and blue images. This is a much more efficient method to perform the interpolation and it produces similar results, and better results in some cases. The key to AH-POCS is to use an accurate result for the green interpolation. For this, the POCS algorithm was used since it produced the best results of the algorithms tested.

(a)

(b)

Figure 77: *Houses* image. (a) Original image. (b) Bilinear interpolation result.

(a)

(b)

Figure 78: *Houses* image. (a) POCS. (b) AH-POCS.

(a) (b)

Figure 79: *Lighthouse* image. (a) Original image. (b) Bilinear interpolation result.

(a) (b)

Figure 80: *Lighthouse* image. (a) POCS. (b) AH-POCS.

(a)

(b)

Figure 81: *Motorcycles* image. (a) Original image. (b) Bilinear interpolation result.

(a)

(b)

Figure 82: *Motorcycles* image. (a) POCS. (b) AH-POCS.

CHAPTER VII

CONCLUSION

7.1 Contributions

This thesis provides an overview of the color filter array interpolation problem, a problem unique to digital cameras where a single sensor array is used to capture a color image. Contributions of this thesis include the following.

- The CFA interpolation problem is presented. The sampling equations for the red, green, and blue sampled images are given in terms of the original signals. This gives a description of the inherent aliasing in the system.

- A mathematical description is presented for the Adams and Hamilton (AH) algorithm and POCS. This description provides an explanation for aliasing in the green image and also, in the case of AH, the red and blue images.

- A method of interpolating the green image using the red and blue images for correction is described. Inspired by the AH algorithm, a set of filters is designed to interpolate the green image and correct the interpolation using information from the red and blue images. This system does not perform a directional interpolation decision, so it is more efficient to implement on a DSP microprocessor architecture.

- Also defined is a second method of interpolating the green image using a filterbank system to cancel aliasing in the interpolated green image. The low-frequency subbands from the green image generate the interpolated green image. The high-frequency subbands from the red and blue images are used to cancel aliasing in the green image. This method performs better than AH, but not as well as POCS. The edge-directed interpolation used by POCS helps this algorithm to perform better in most images.

- A method of interpolating the red and blue images that uses the green image to cancel

aliasing in the interpolated red and blue images is presented. The high frequencies are isolated using a highpass filter. The horizontal high frequencies are modulated horizontally to estimate aliasing due to the horizontal components and the vertical high frequencies are modulated vertically to estimate aliasing from the vertical components. The main drawback from this method occurs when the original green image contains components with a spatial frequency outside the diamond-shaped Nyquist region. In this case, horizontal components are confused for vertical components and vertical components are confused for horizontal components. This causes the aliasing to be amplified rather than cancelled.

- Using the same idea of cancelling aliasing, a second method was defined, but this method uses the interpolated green image to obtain the high-frequency information. A set of correction filters is defined for the interpolation process. One filter interpolates the red or blue image from the red or blue samples. The correction filters are applied to the green image to isolate the high-frequency components. This interpolation algorithm is dependent on having a good green interpolated image. It was found that the POCS algorithm provides the best results for this purpose. The algorithm, because it uses only linear filters, is efficient to implement.

This research shows that of the methods studied, POCS performs best for the green image. Aliasing is minimized in the output image and the mean square error is lowest in the POCS result images. The filterbank method for interpolating the green image was developed to match the results of POCS by copying the aliasing reduction in the POCS green interpolation. However, using a non-adaptive method prevents the filterbank method from equalling the results of POCS. High-frequency components along the horizontal and vertical axes near the diamond-shaped Nyquist region are slightly attenuated by the filterbank interpolation, which reduces sharpness in the output image. Based on the chirp example image in Figure 60, both POCS and the filterbank method reduce aliasing by the same amount.

For the red and blue interpolation, AH-POCS performs as well as POCS, but is much simpler to implement. POCS uses an iterative algorithm to interpolate the red and blue images. AH-POCS uses only 3x3 interpolation and correction filters. A method to design the correction filters based on the red and blue interpolation filter was developed to generalize the red and blue interpolation algorithm by Adams and Hamilton. Allowing larger filters provides a more selective frequency response, but this had little effect on the mean square error of the output images. Therefore, it is recommended to only use the 3x3 correction system by Adams and Hamilton.

According to this research, the best method of CFA interpolation mixes two existing methods. The POCS algorithm for the green interpolation is fairly efficient and produces the best results. Using this green image, the simple correction system, AH-POCS, provides the best red and blue image results.

7.2 Future Research

Color filter array interpolation is an active area of research. There are several angles that remain to be thoroughly researched.

- CFA interpolation is one step in a long chain of image processing inside the digital camera. Few researchers describe the interaction between the interpolation step and the other processing steps. An interesting area that has been briefly described [6] is the competition between CFA interpolation and image compression. The interpolation step tries to keep as much information as possible, but then the compression step decides what information can be rejected. Interaction between these two processing steps could provide a more efficient and possibly better processing chain.

- A new type of sensor has been invented for use in digital still cameras [21, 19]. This sensor uses the light absorption properties of silicon to measure the amount of light in the red, green, and blue portions at different depths in the chip . This allows a camera to measure a red, green, and blue value for every pixel of the sensor. This changes the current structure of the image processing chain in cameras, eliminating the need for

color filter array interpolation. Research to understand the effects of using this type of sensor in cameras needs to be performed.

- A current trend with consumer cameras, especially more expensive models, is to provide the user with the raw sensor data recorded by the image sensor. This allows the user to do post-processing off-line, allowing more complex algorithms to be used. This removes a constraint on current algorithms. Do algorithms exist that are too complex for real-time use, but can be used in off-line systems where time isn't a critical issue?

REFERENCES

[1] Adams, Jr., J. E., "Design of practical color filter array interpolation algorithms for digital cameras," in *Proc. SPIE*, vol. 3028, pp. 117–125, 1997.

[2] Adams, Jr., J. E., "Design of practical color filter array interpolation algorithms for digital cameras, part 2," in *Proc. Int. Conf. Image Proc.*, pp. 488–492, 1998.

[3] Adams, Jr., J. E. and Hamilton, Jr., J. F., "Adaptive color plan interpolation in single sensor color electronic camera." U.S. Patent 5,506,619, 1996.

[4] Adams, Jr., J. E. and Hamilton, Jr., J. F., "Adaptive color plane interpolation in single sensor color electronic camera." U.S. Patent 5,652,621, 1997.

[5] Adams, J., Parulski, K., and Spaulding, K., "Color processing in digital cameras," *IEEE Micro*, vol. 18, pp. 20–30, Nov-Dec 1998.

[6] Baharav, Z. and Kakarala, R., "Compression aware demosaicing methods," in *Proceedings of SPIE*, vol. 4667, pp. 149–155, 2002.

[7] Bayer, B. E., "Color imaging array." U.S. Patent 3,971,065, 1976.

[8] Chang, E., Cheung, S., and Pan, D., "Color filter array recovery using a threshold-based variable number of gradients," in *Proc. SPIE*, vol. 3650, pp. 36–43, 1999.

[9] Cok, D. R., "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal.." U.S. Patent 4,642,678, 1987.

[10] Daubechies, I., *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[11] Dudgeon, D. E. and Mersereau, R. M., *Multidimensional Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1984.

[12] Gunturk, B. K., Altunbasak, Y., and Mersereau, R. M., "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, pp. 997–1013, Sept. 2002.

[13] Hamilton, Jr., J. F. and Adams, Jr., J. E., "Adaptive color plan interpolation in single sensor color electronic camera." U.S. Patent 5,629,734, 1997.

[14] Hou, H. S. and Andrews, H. C., "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 508–517, December 1978.

[15] Hu, J. V. and Rabiner, L. R., "Design techniques for two-dimensional digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-20, pp. 249–257, October 1972.

[16] Keys, R. G., "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 1153–1160, December 1981.

[17] Kimmel, R., "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Proc.*, vol. 8, pp. 1221–1229, Sept. 1999.

[18] Lee, B., Kim, J., and Lee, C., "High quality image interpolation for color filter arrays," in *Proc. Int. Conf. Systems, Man, and Cybernetics*, vol. 2, pp. 1547–1550, 2000.

[19] Lyon, R. F. and Hubel, P. M., "Eyeing the camera: Into the next century," in *IS&T/SID Tenth Color Imaging Conference*, (Scottsdale, AZ), pp. 349–355, 2002.

[20] Marimont, D. H. and Wandell, B. A., "Matching color images: the effects of axial chromatic aberration," *Journal of Optical Society of America. A*, vol. 11, pp. 3113–3122, December 1994.

[21] Merrill, R. B., "Color separation in an active pixel cell imaging array using a triple-well structure." U.S. Patent 5,965,875, 1989.

[22] Pratt, W. K., *Digital Image Processing*. New York: John Wiley and Sons, Inc., 1991.

[23] Rabiner, L. R., "Linear program design of finite impulse response (FIR) digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-20, pp. 280–288, October 1972.

[24] The Mathworks, *Matlab Wavelet Toolbox Version 2.0*. 2000.

[25] Unser, M., "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, pp. 22–38, November 1999.

[26] Wandell, B. A., *Foundations of Vision*. Sunderland, MA: Sinauer Associates, Inc., 1995.

[27] Wyszecki, G. and Stiles, W. S., *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd ed.* New York: John Wiley and Sons, Inc., 1982.

VITA

John Glotzbach was born on August 27, 1976 in Lafayette, Indiana. He graduated from Benton Central High School in Oxford, Indiana in 1993. From there, he attended Purdue University and received a Bachelor of Science in Electrical Engineering in May 1998. During his time at Purdue, John was a coop student at Thomson Consumer Electronics in Indianapolis. In September 1998, he moved to Atlanta, Georgia to enroll at Georgia Institute of Technology. He received a Master of Science in Electrical and Computer Engineering in May 2000 and continued his graduate studies at Georgia Tech to work on a PhD degree in Electrical and Computer Engineering. John worked for Texas Instruments, Inc. in Dallas for two summers during this time and will be returning to Texas Instruments at the completion of his graduate studies. John is a member of Eta Kappa Nu and the IEEE Signal Processing Society.